



A paradigm shift

Suchita Doshi

 suchitadoshi

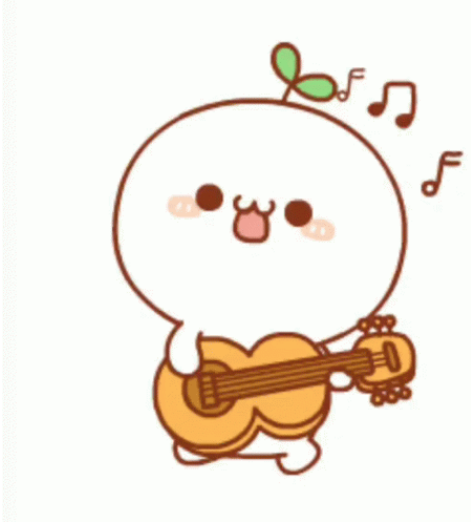
 suchitadoshi1987

 suchita009

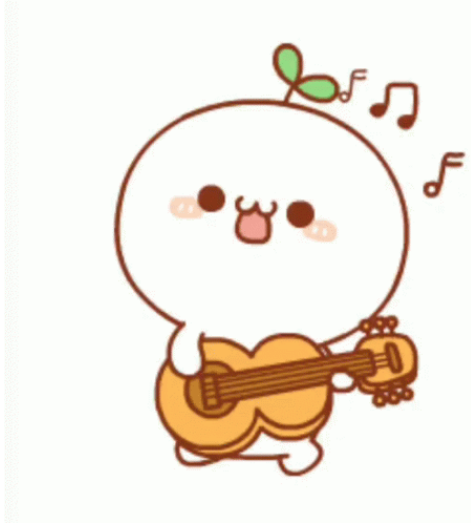
 suchita009

When I am away from my computer...

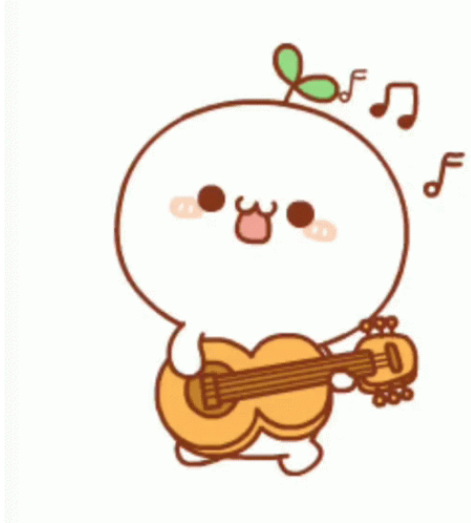
When I am away from my computer...



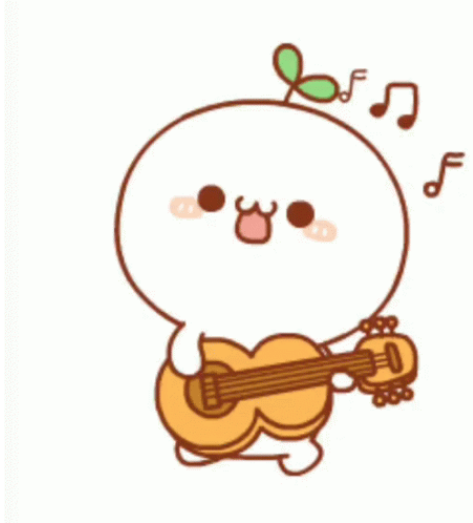
When I am away from my computer...



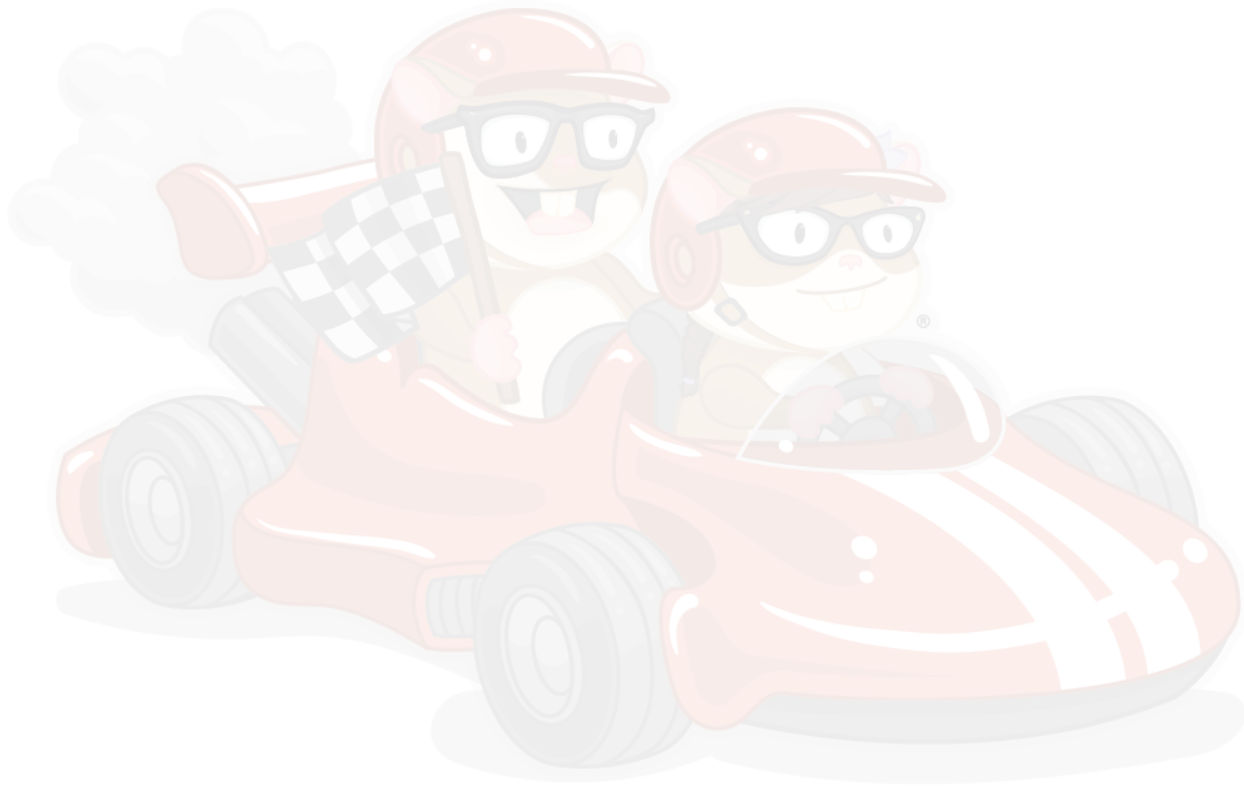
When I am away from my computer...



When I am away from my computer...



Agenda

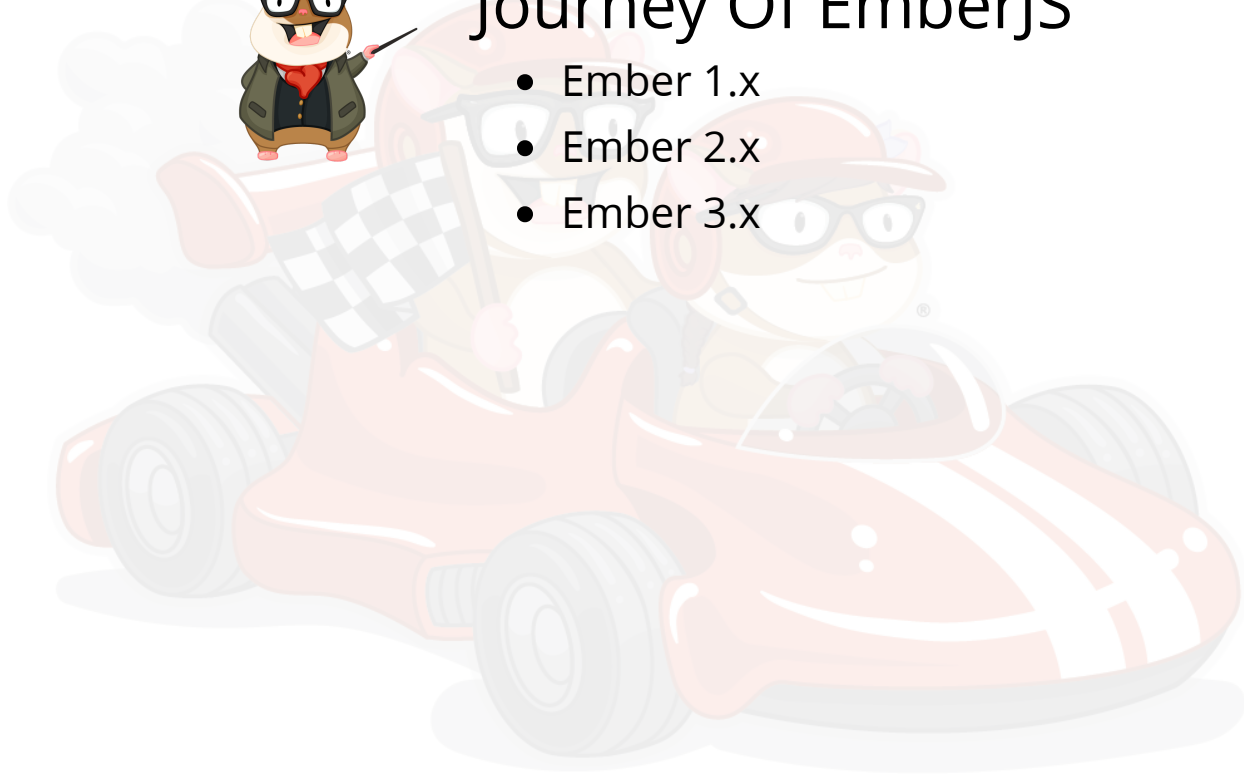


Agenda



Journey Of EmberJS

- Ember 1.x
- Ember 2.x
- Ember 3.x



Agenda



Journey Of EmberJS



Ember Octane

- Native Classes
- Glimmer Components
- Templates in Octane
- Tracked Properties
- Modifiers & Decorators

Agenda



Journey Of EmberJS



Ember Octane



Classic vs Octane Epic
comparison

Agenda



Journey Of EmberJS



Ember Octane



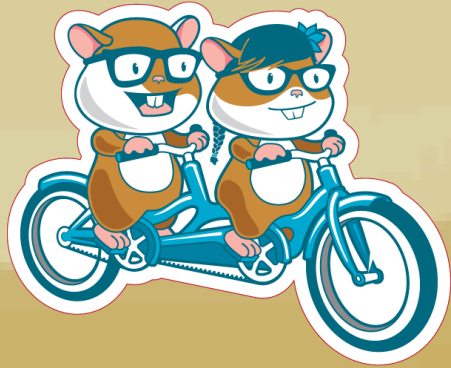
Classic vs Octane Epic
comparison



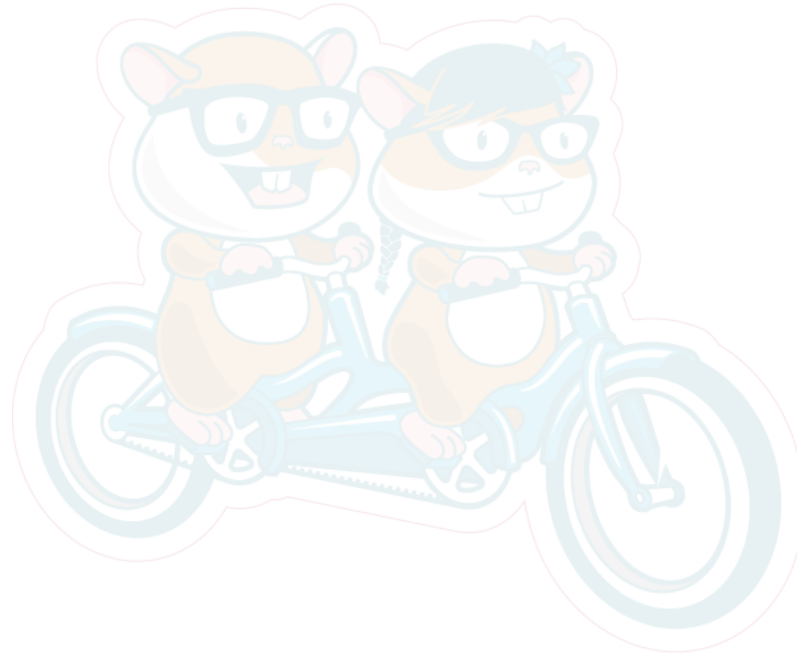
Migration tools &
References

The
JOURNEY
Of

ember®

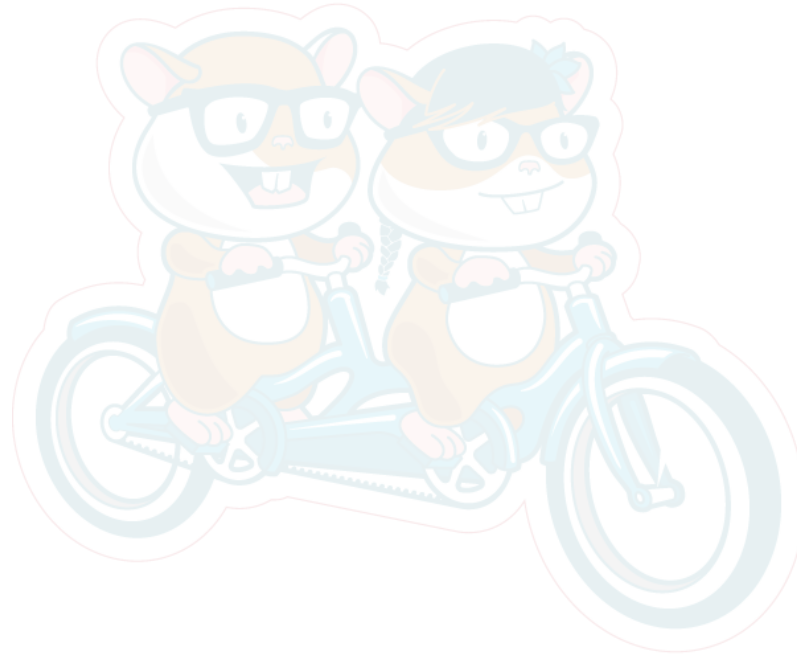


Ember 1.x



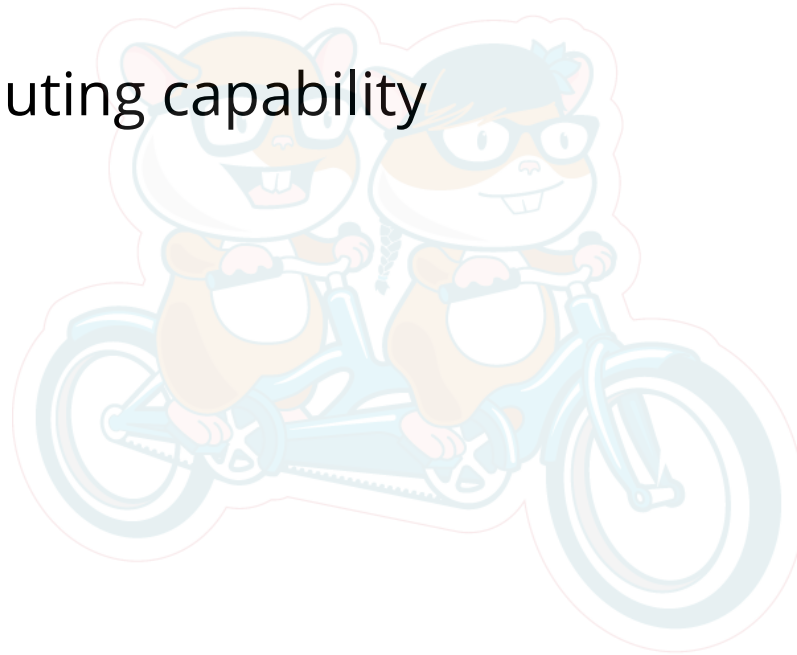
Ember 1.x

- Convention over configuration (A new mental model)



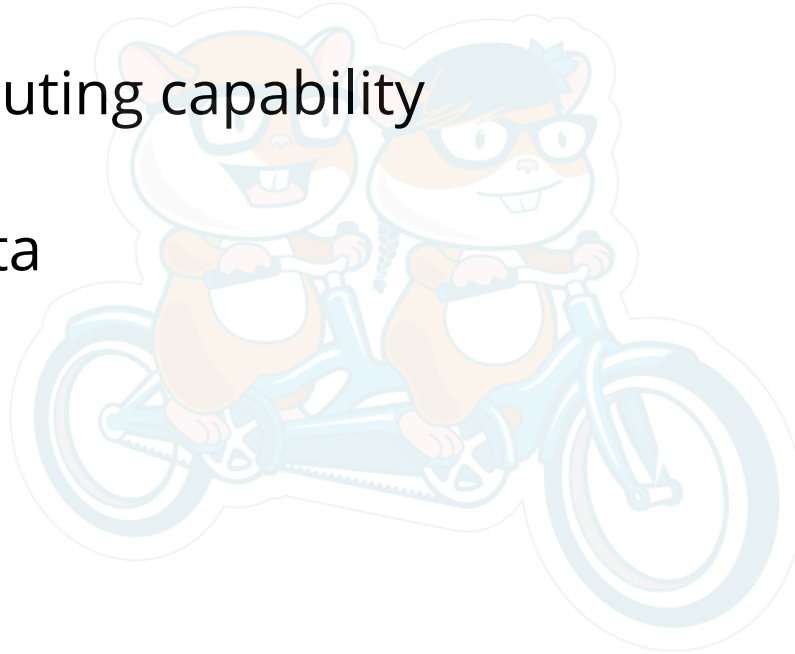
Ember 1.x

- Convention over configuration (A new mental model)
- Built-in Routing capability



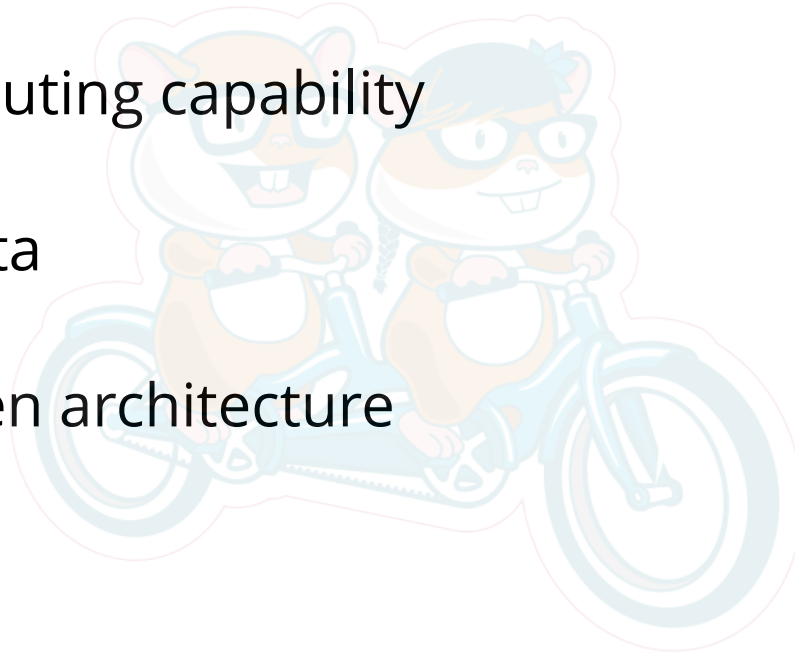
Ember 1.x

- Convention over configuration (A new mental model)
- Built-in Routing capability
- Ember-data

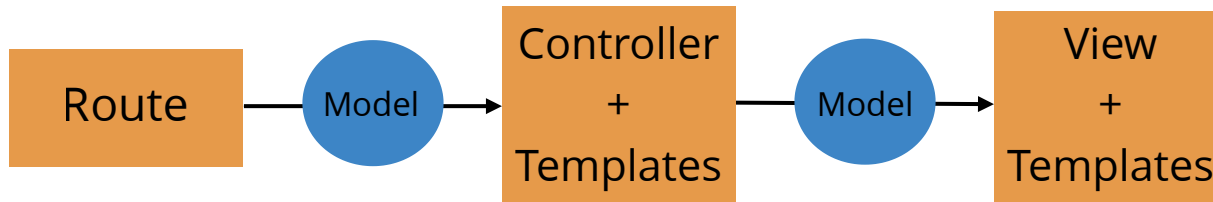


Ember 1.x

- Convention over configuration (A new mental model)
- Built-in Routing capability
- Ember-data
- View Driven architecture



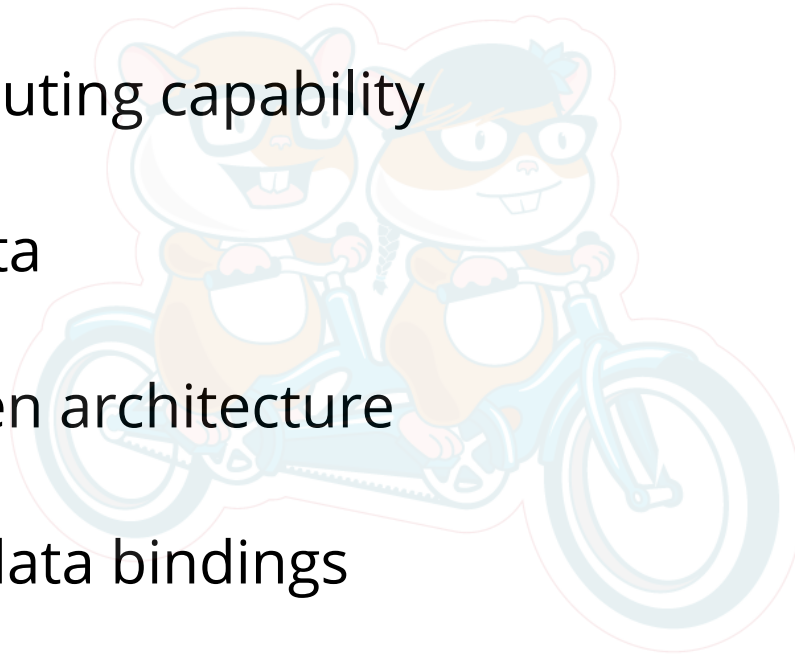
Ember 1.x



```
1 export default Ember.View.extend({
2   classNameBindings: ['isActive'],
3   isActive: true,
4   firstName: 'John',
5   lastName: 'Doe',
6
7   // Computed Property
8   fullName: function() {
9     return this.get('firstName') + ' ' + this.get('lastName');
10  }.property('firstName', 'lastName'),
11
12  fullNameChanged: function() {
13    // deal with the change
14  }.observes('fullName')
15 });
```

Ember 1.x

- Convention over configuration (A new mental model)
- Built-in Routing capability
- Ember-data
- View Driven architecture
- Two way data bindings

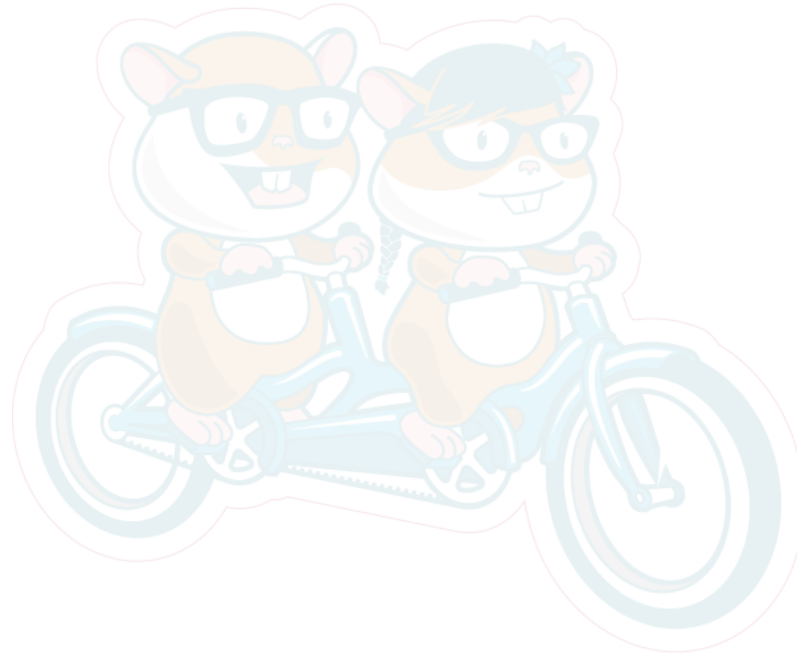


Ember 1.x

- Convention over configuration (A new mental model)
- Built-in Routing capability
- Ember-data
- View Driven architecture
- Two way data bindings
- Attribute bindings using `{{bind-attr}}`

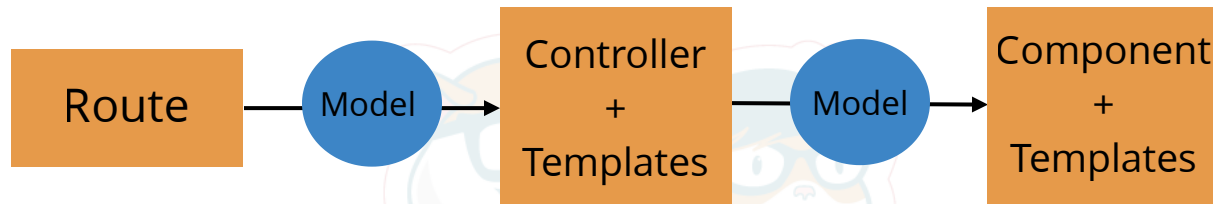
```
1 <div {{bind-attr title=post.popup}}></div>
```

Ember 2.x



Ember 2.x

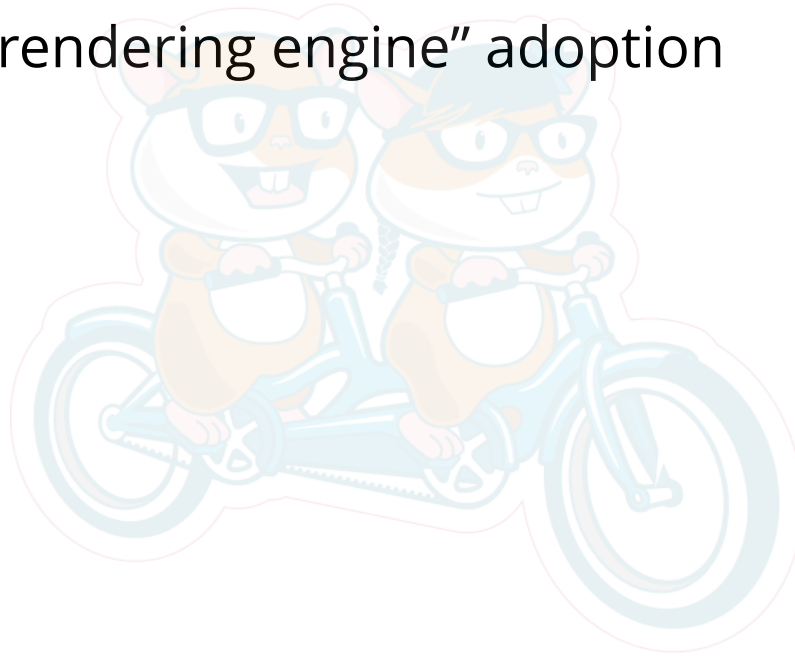
- Component driven



```
1 export default Ember.Component.extend({
2   classNameBindings: ['isActive'],
3   isActive: true,
4   firstName: 'John',
5   lastName: 'Doe',
6
7   // Computed Property
8   fullName: Ember.computed('firstName', 'lastName', function() {
9     return this.get('firstName') + ' ' + this.get('lastName');
10  }),
11
12  fullNameChanged: Ember.observer('fullName', function() {
13    // deal with the change
14  })
15 });
```

Ember 2.x

- Component driven
- “Glimmer rendering engine” adoption



Ember 2.x

- Component driven
- “Glimmer rendering engine” adoption
- Better binding with properties ~~{{bind-attr}}~~

```
// v1.x
```

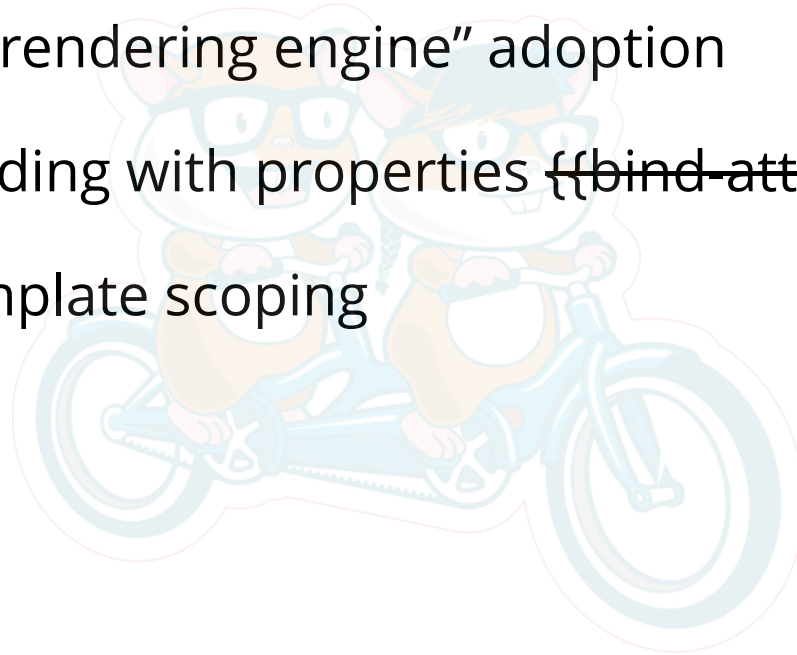
```
1 <div {{bind-attr title=post.popup}}></div>
```

```
// v2.x
```

```
1 <div title="{{post.popup}}"></div>
```


Ember 2.x

- Component driven
- “Glimmer rendering engine” adoption
- Better binding with properties `{{bind-attr}}`
- Better template scoping



Ember 2.x

- Component driven
- “Glimmer rendering engine” adoption
- Better binding with properties ~~{{bind-attr}}~~
- Better template scoping

// v1.x

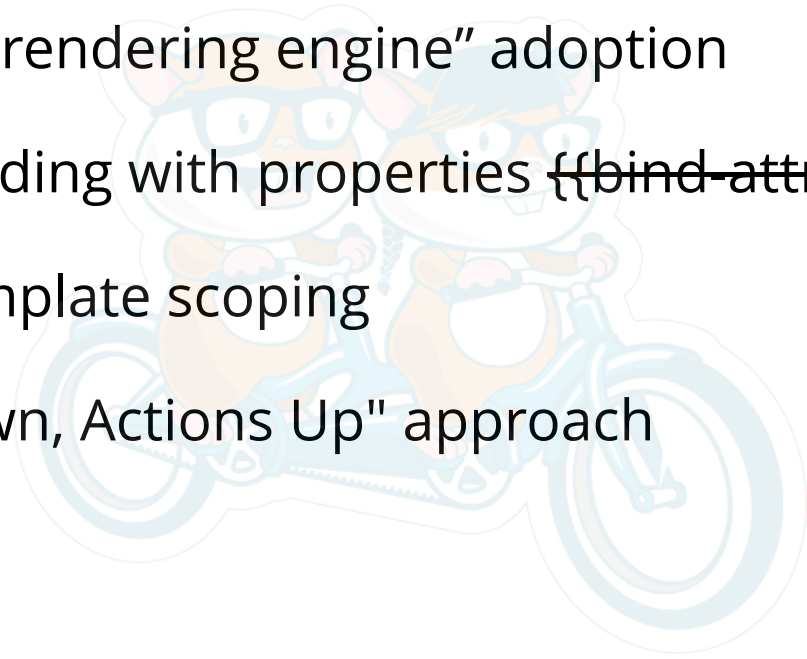
```
1  {{{!-- 1.x version --}}}
2  {{#each post in posts}}
3    {{{!-- `post` references
4    the current iteration --}}}
5  {{/each}}
6
7  {{#each posts}}
8    {{{!-- the outer context
9    is no longer accessible --}}}
10 {{/each}}
11
```

// v2.x

```
1  {{{!-- 2.x version --}}}
2  {{#each posts as |post|}}
3    {{{!-- `post` references
4    the current iteration --}}}
5    <p>{{post.id}} {{post.title}}</p>
6  {{/each}}
7
```

Ember 2.x

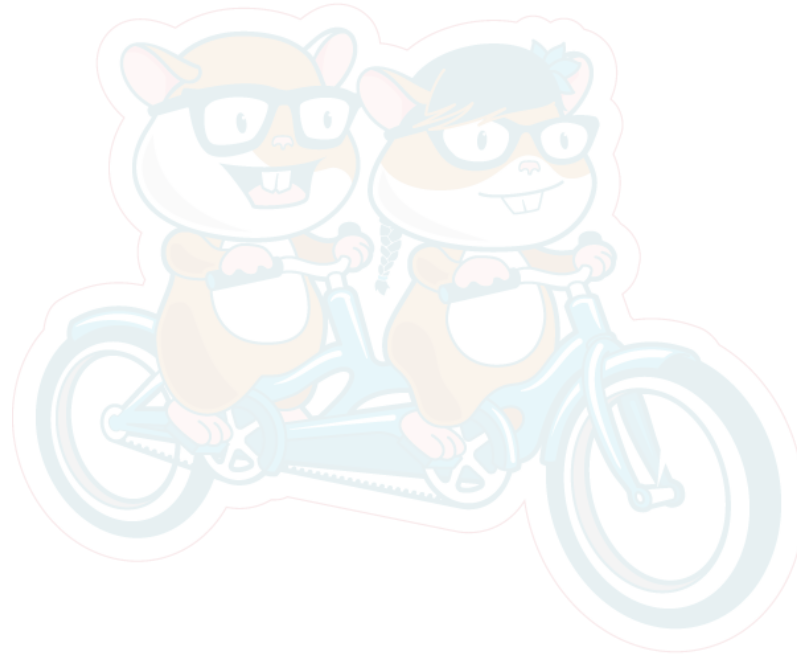
- Component driven
- “Glimmer rendering engine” adoption
- Better binding with properties `{{bind-attr}}`
- Better template scoping
- "Data Down, Actions Up" approach



Ember 2.x

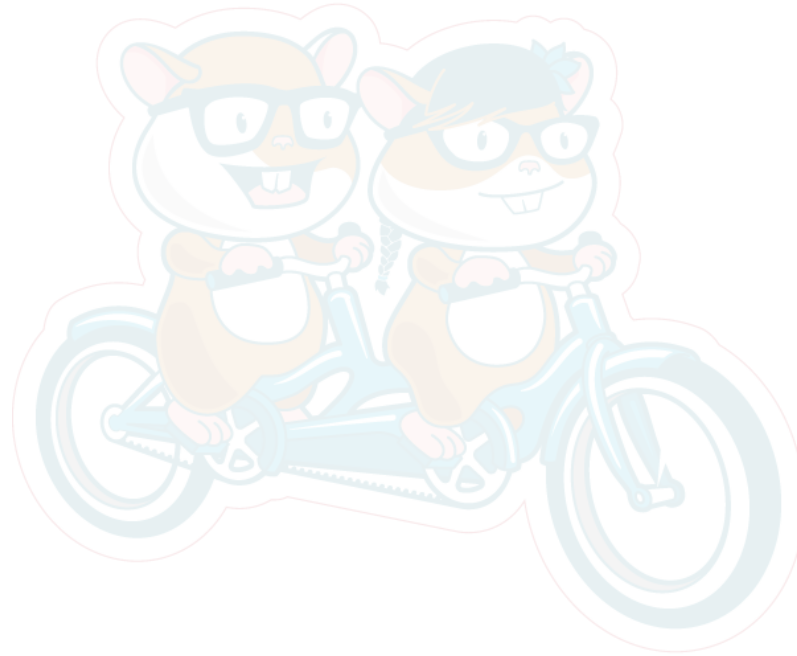
- Component driven
- “Glimmer rendering engine” adoption
- Better binding with properties `{{bind-attr}}`
- Better template scoping
- "Data Down, Actions Up" approach
- Roadmap for a lot of further improvements:
 - HTML syntax for component invocation
 - Routes to drive Components approach

Ember 3.x (Road to Octane)



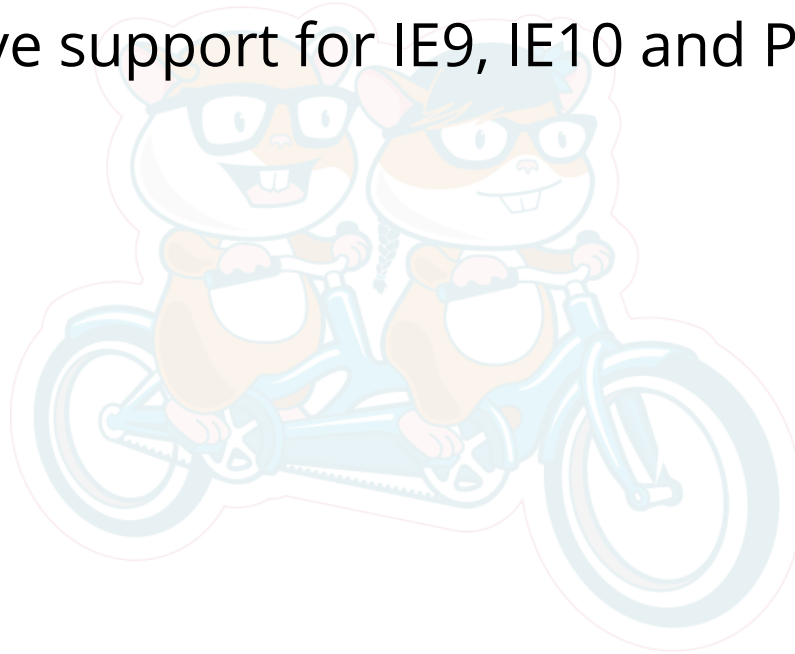
Ember 3.x (Road to Octane)

- Cleanup Cleanup Cleanup



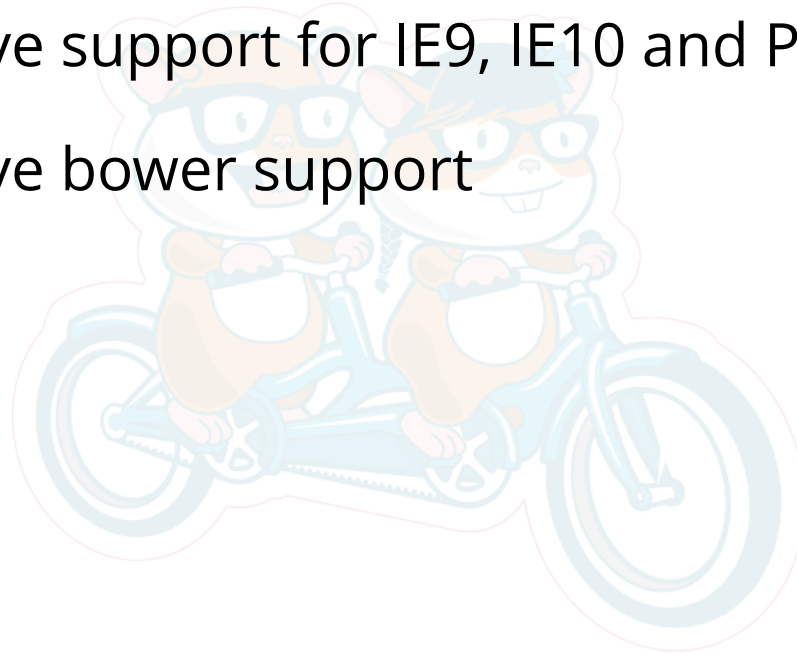
Ember 3.x (Road to Octane)

- Cleanup Cleanup Cleanup
- Remove support for IE9, IE10 and PhantomJS.



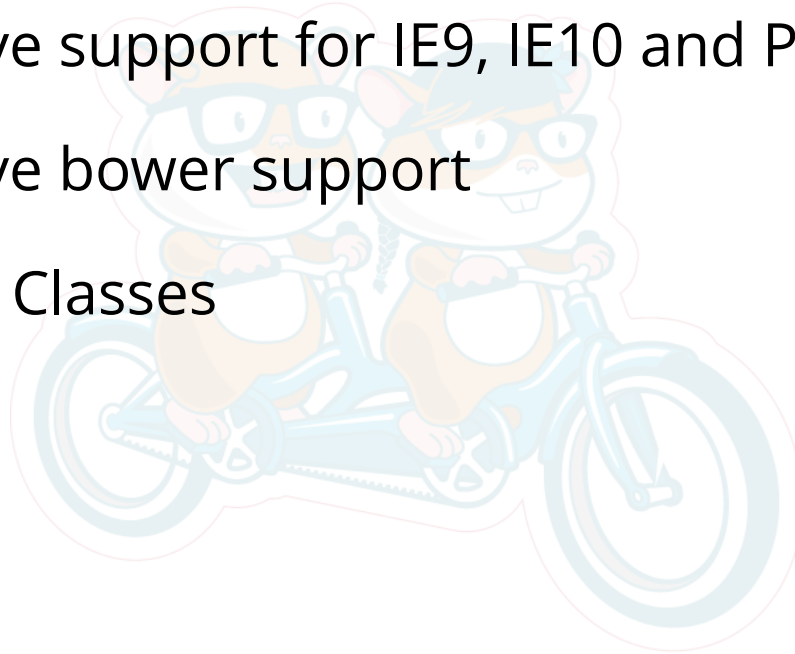
Ember 3.x (Road to Octane)

- Cleanup Cleanup Cleanup
- Remove support for IE9, IE10 and PhantomJS.
- Remove bower support



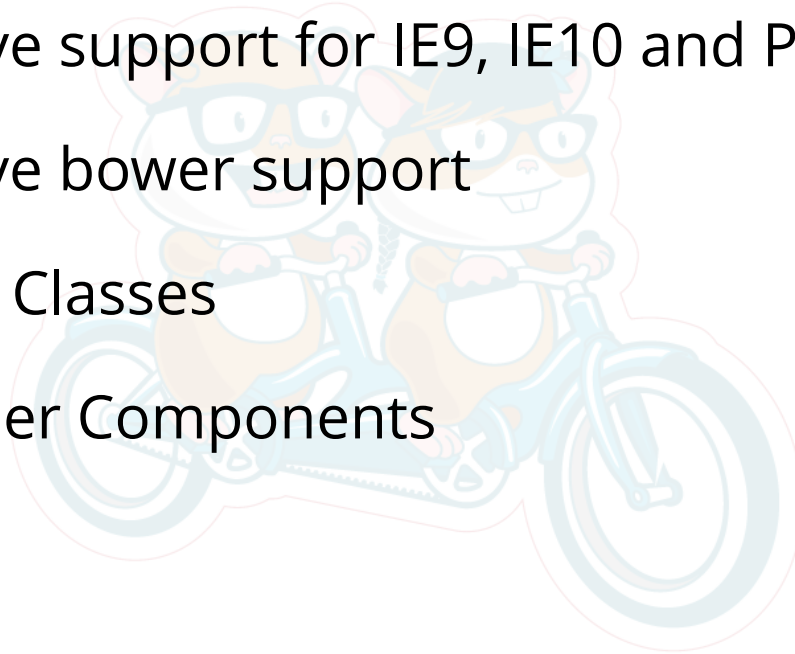
Ember 3.x (Road to Octane)

- Cleanup Cleanup Cleanup
- Remove support for IE9, IE10 and PhantomJS.
- Remove bower support
- Native Classes



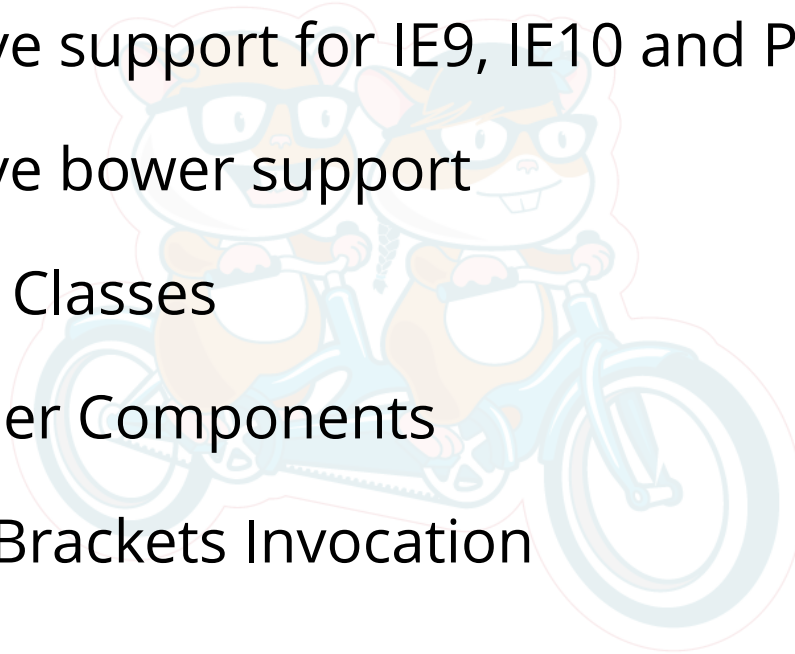
Ember 3.x (Road to Octane)

- Cleanup Cleanup Cleanup
- Remove support for IE9, IE10 and PhantomJS.
- Remove bower support
- Native Classes
- Glimmer Components



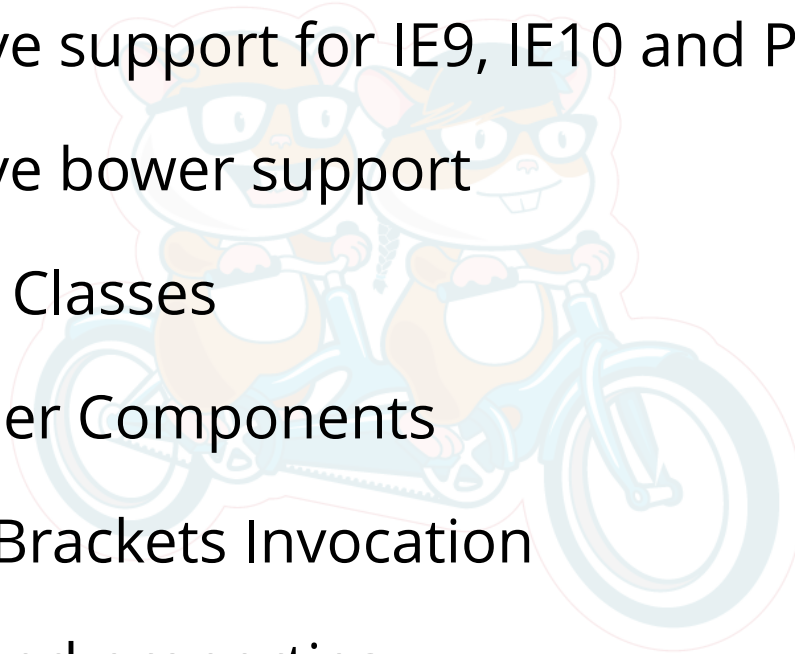
Ember 3.x (Road to Octane)

- Cleanup Cleanup Cleanup
- Remove support for IE9, IE10 and PhantomJS.
- Remove bower support
- Native Classes
- Glimmer Components
- Angle Brackets Invocation



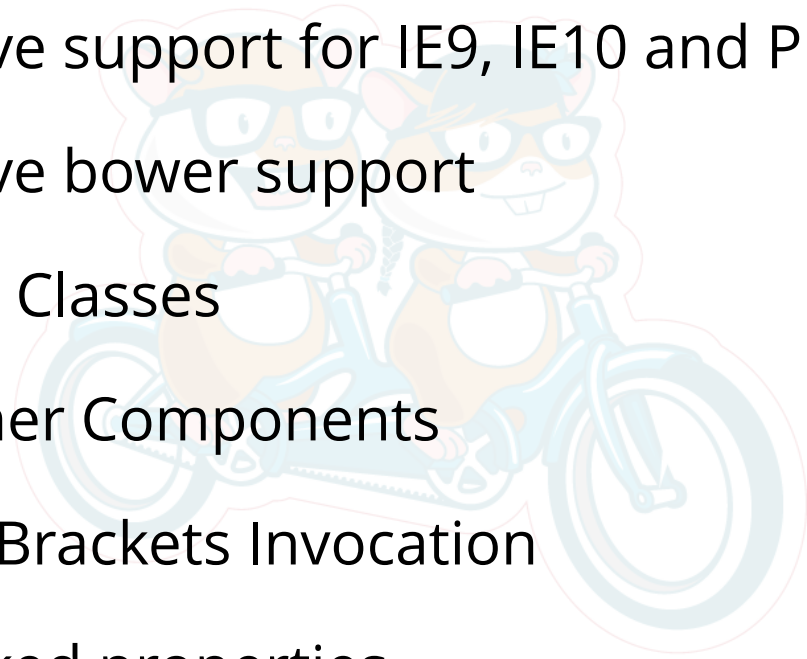
Ember 3.x (Road to Octane)

- Cleanup Cleanup Cleanup
- Remove support for IE9, IE10 and PhantomJS.
- Remove bower support
- Native Classes
- Glimmer Components
- Angle Brackets Invocation
- @tracked properties



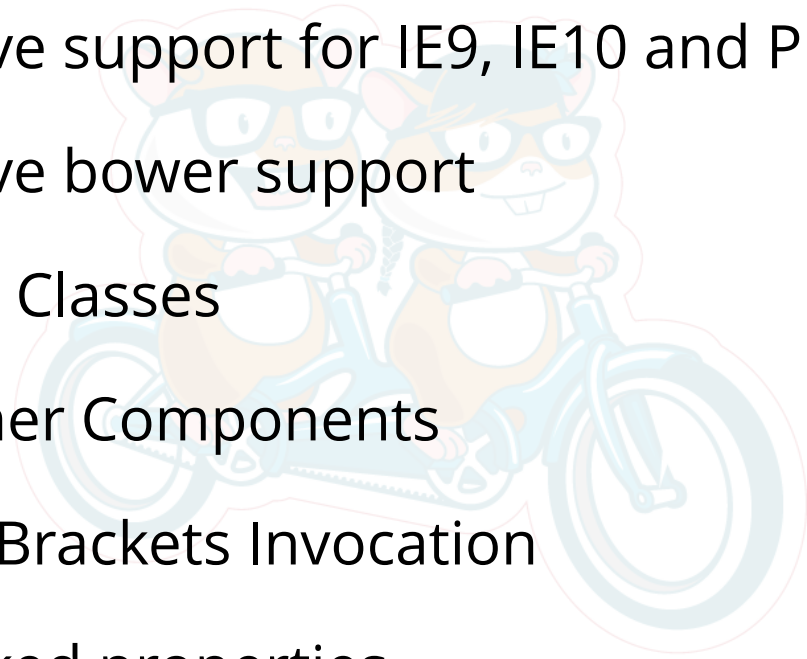
Ember 3.x (Road to Octane)

- Cleanup Cleanup Cleanup
- Remove support for IE9, IE10 and PhantomJS.
- Remove bower support
- Native Classes
- Glimmer Components
- Angle Brackets Invocation
- @tracked properties
- Modifiers & Decorators



Ember 3.x (Road to Octane)

- Cleanup Cleanup Cleanup
- Remove support for IE9, IE10 and PhantomJS.
- Remove bower support
- Native Classes
- Glimmer Components
- Angle Brackets Invocation
- @tracked properties
- Modifiers & Decorators
- Lots of documentation



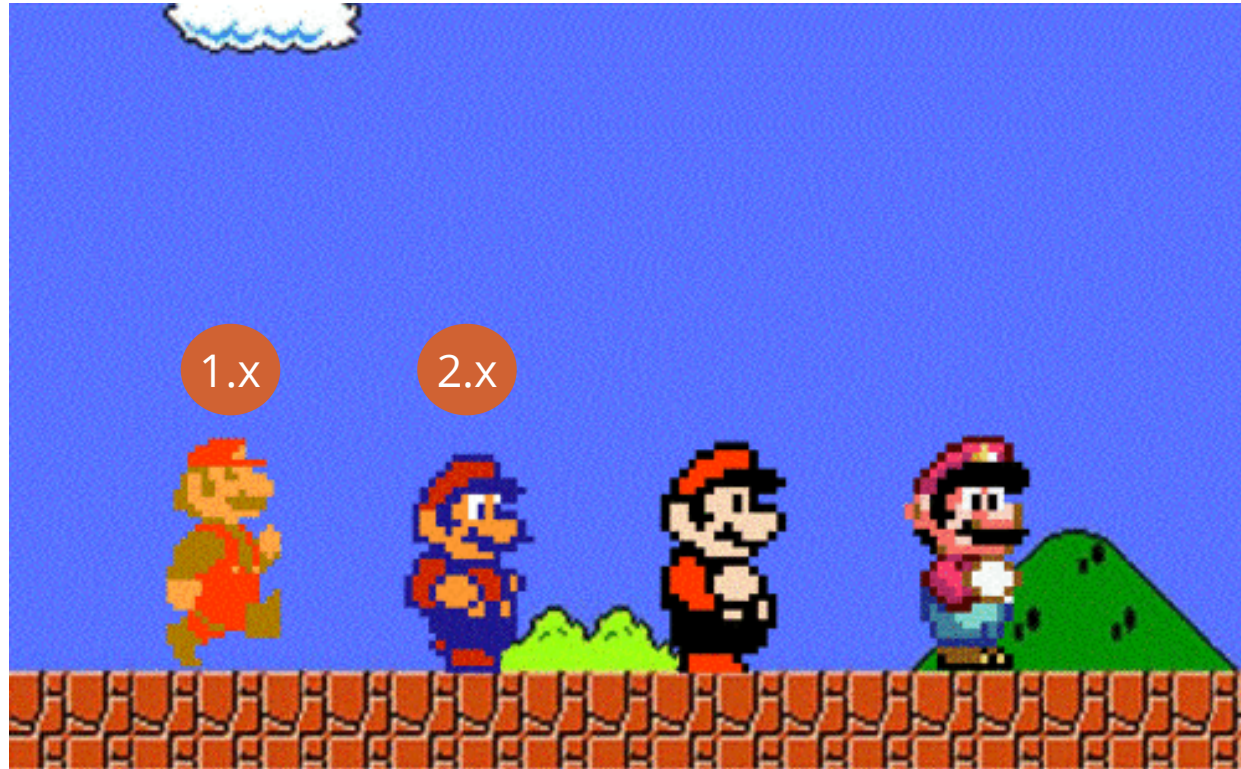
Evolution of EmberJS



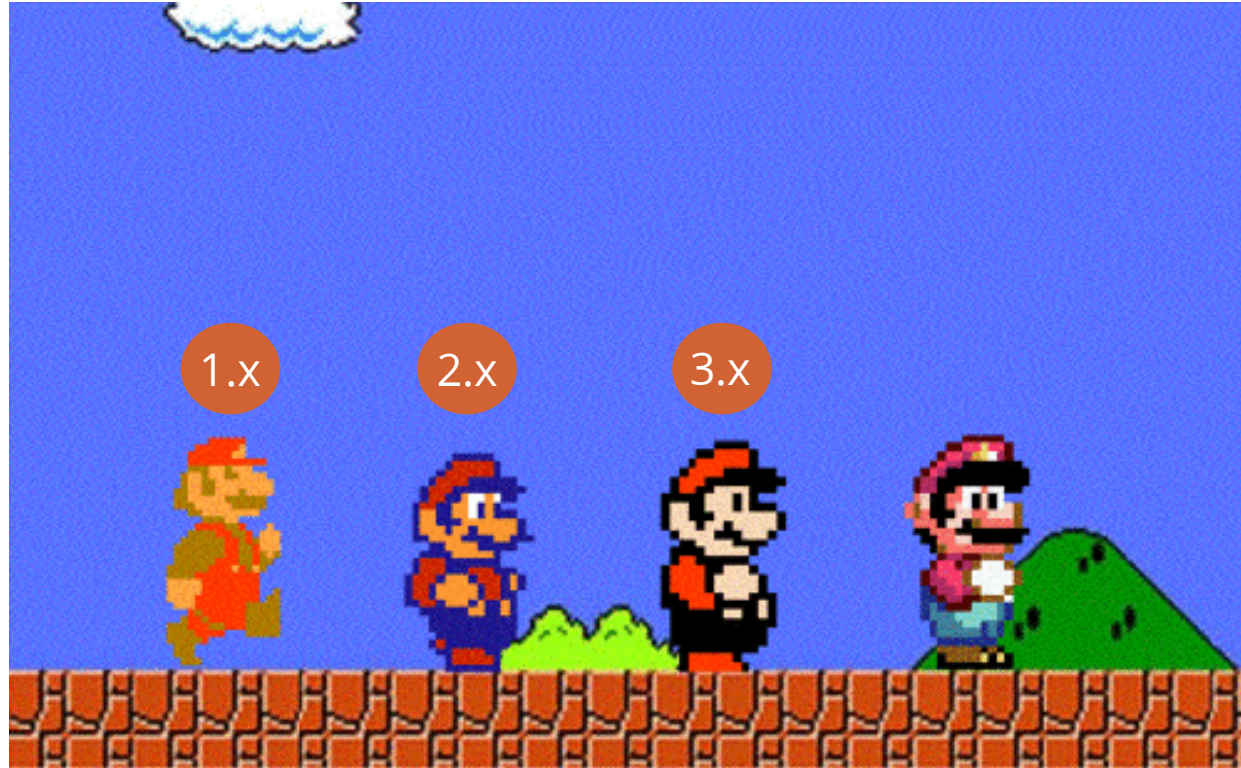
Evolution of EmberJS



Evolution of EmberJS



Evolution of EmberJS



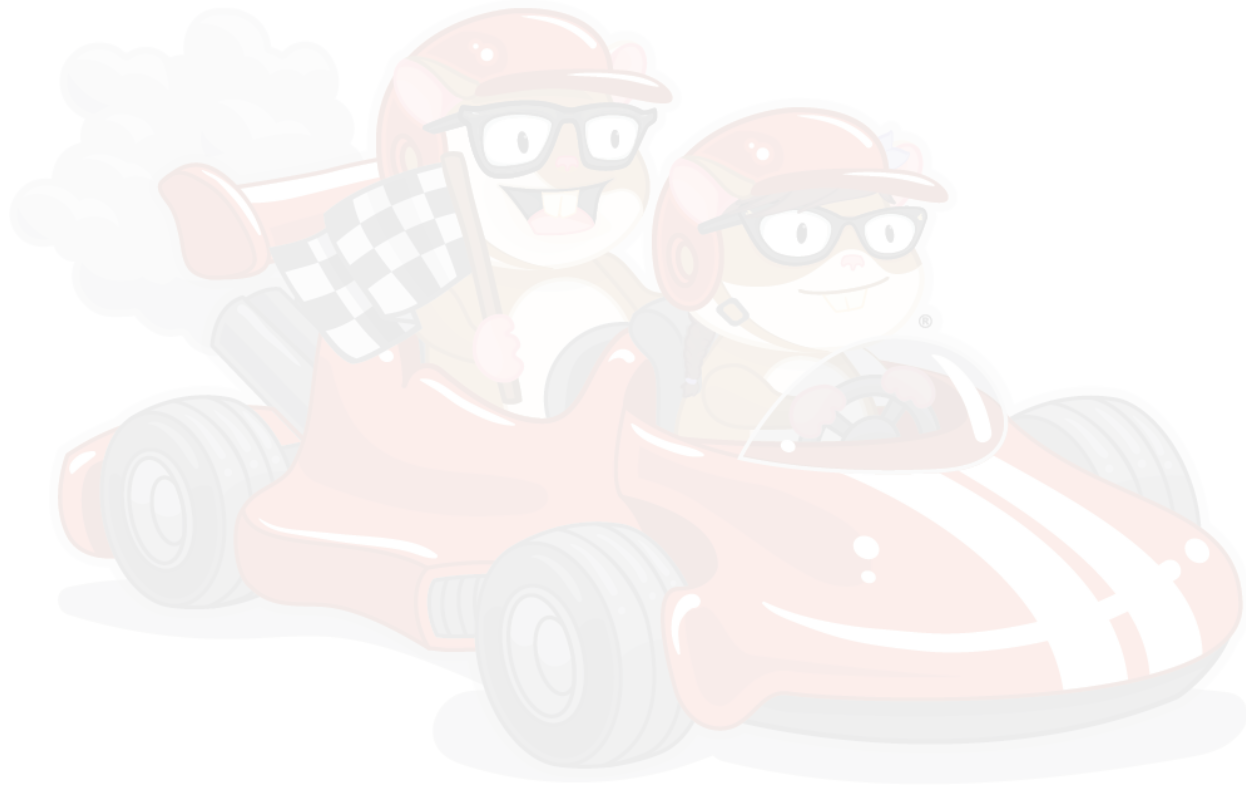
Evolution of EmberJS



ember[®] **OCTANE**

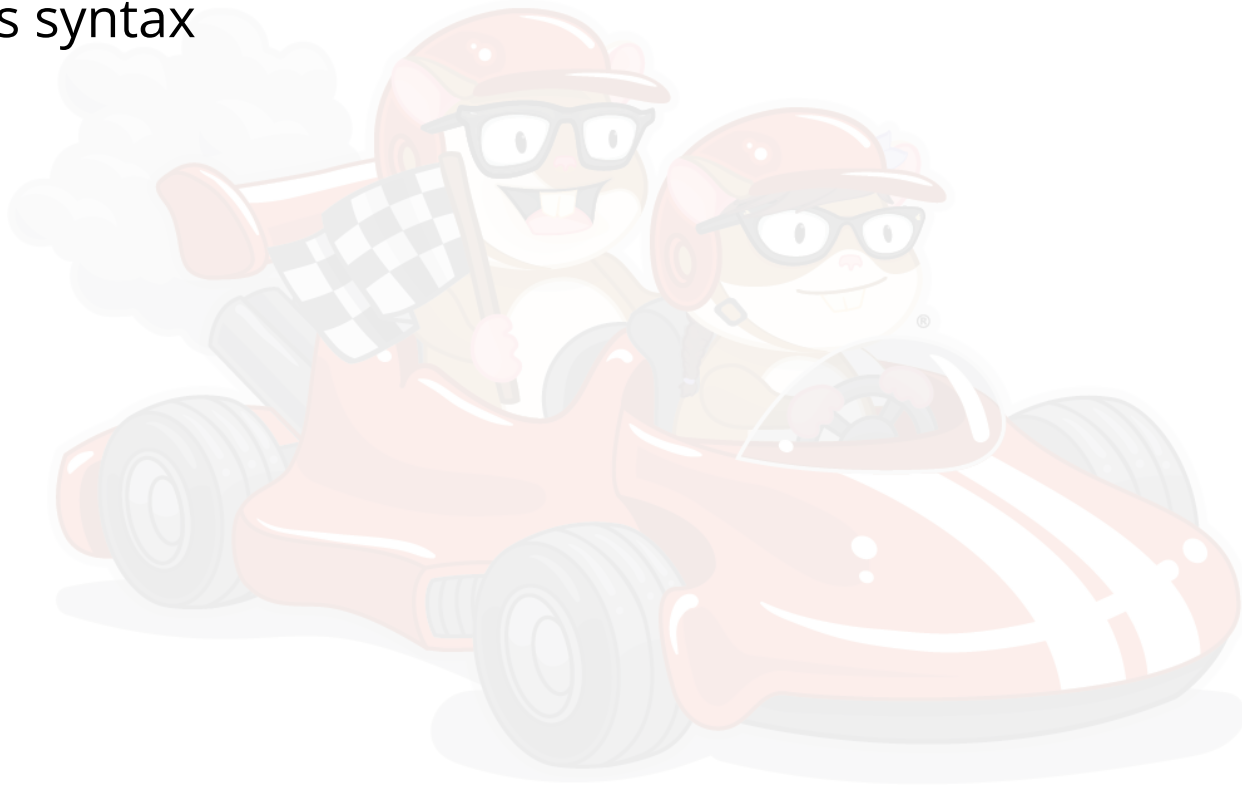


Native Classes



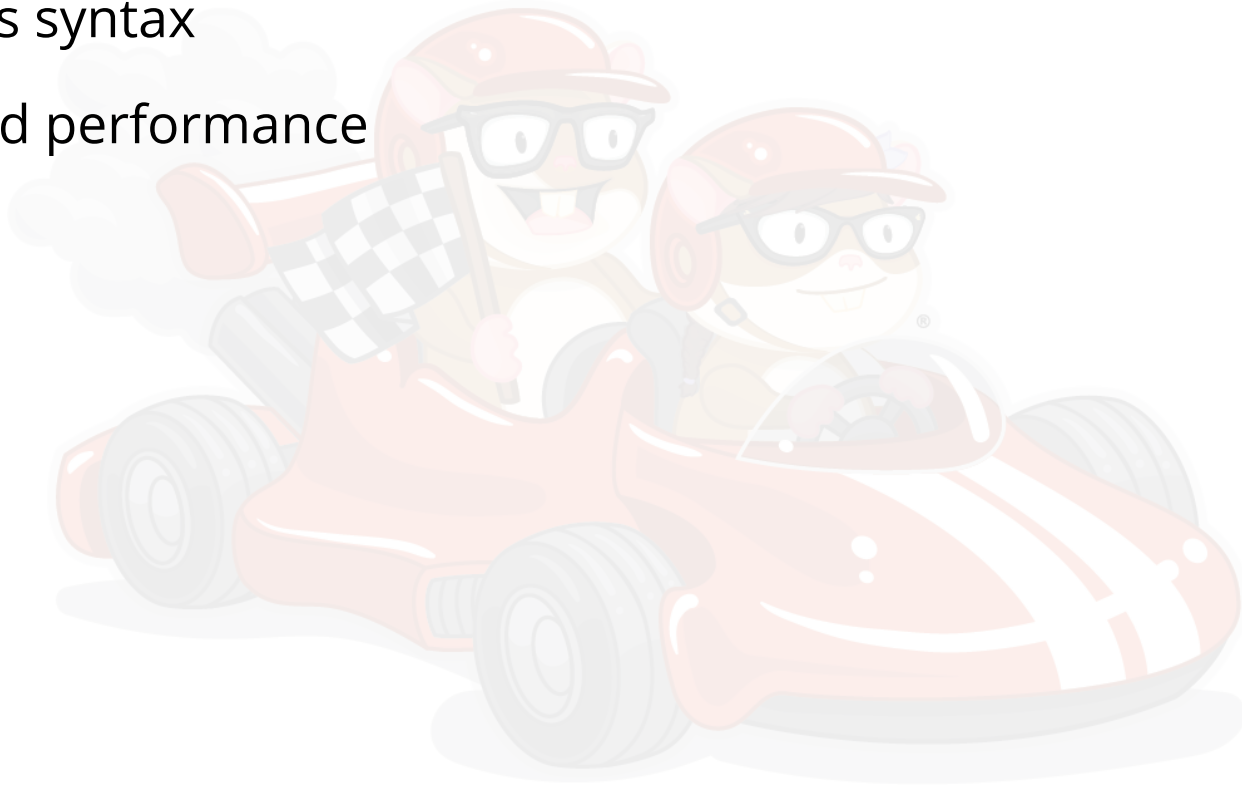
Native Classes

- ES6 Class syntax



Native Classes

- ES6 Class syntax
- Increased performance



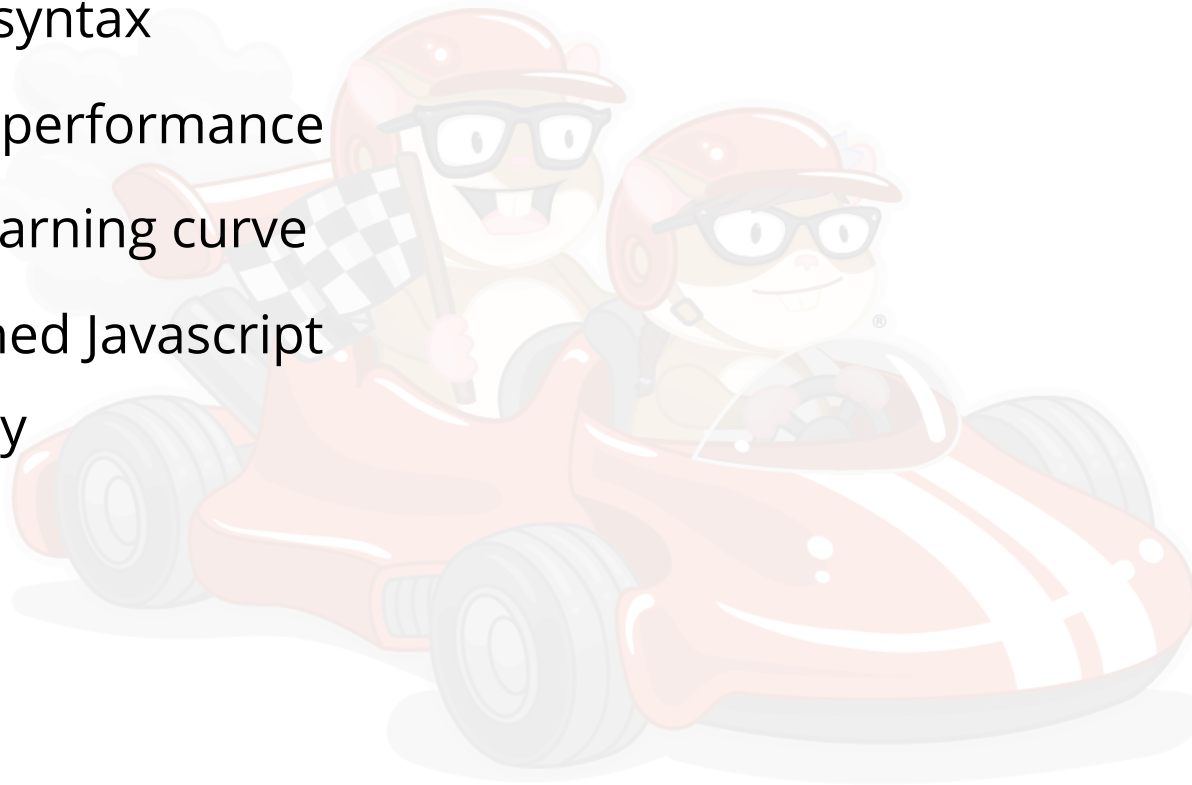
Native Classes

- ES6 Class syntax
- Increased performance
- Smooth learning curve



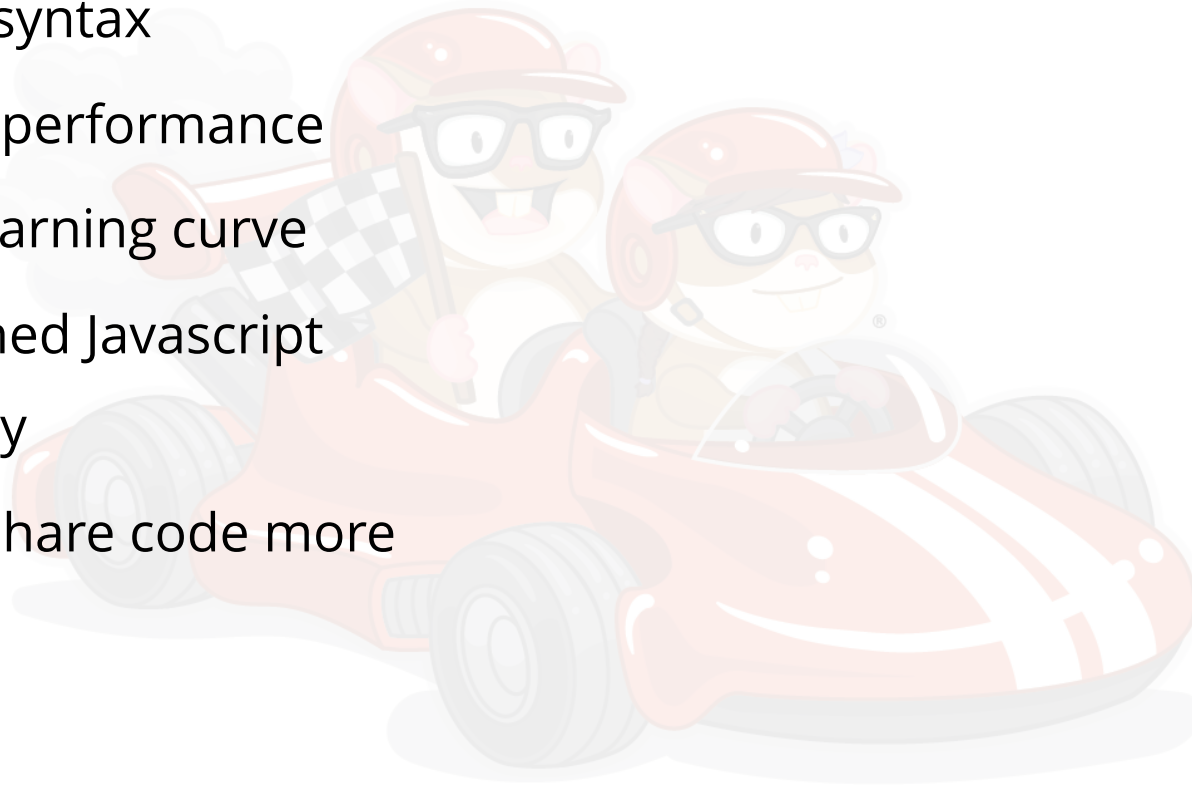
Native Classes

- ES6 Class syntax
- Increased performance
- Smooth learning curve
- More aligned Javascript community



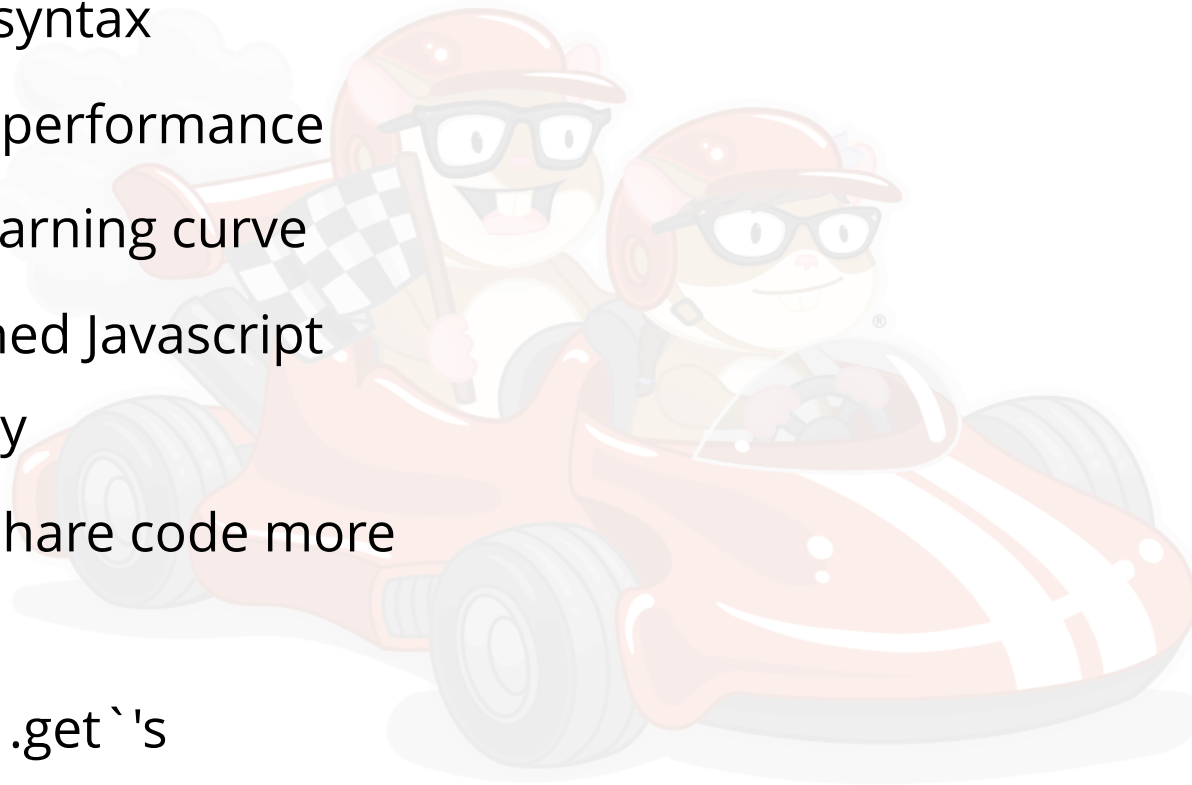
Native Classes

- ES6 Class syntax
- Increased performance
- Smooth learning curve
- More aligned Javascript community
- Ability to share code more easily



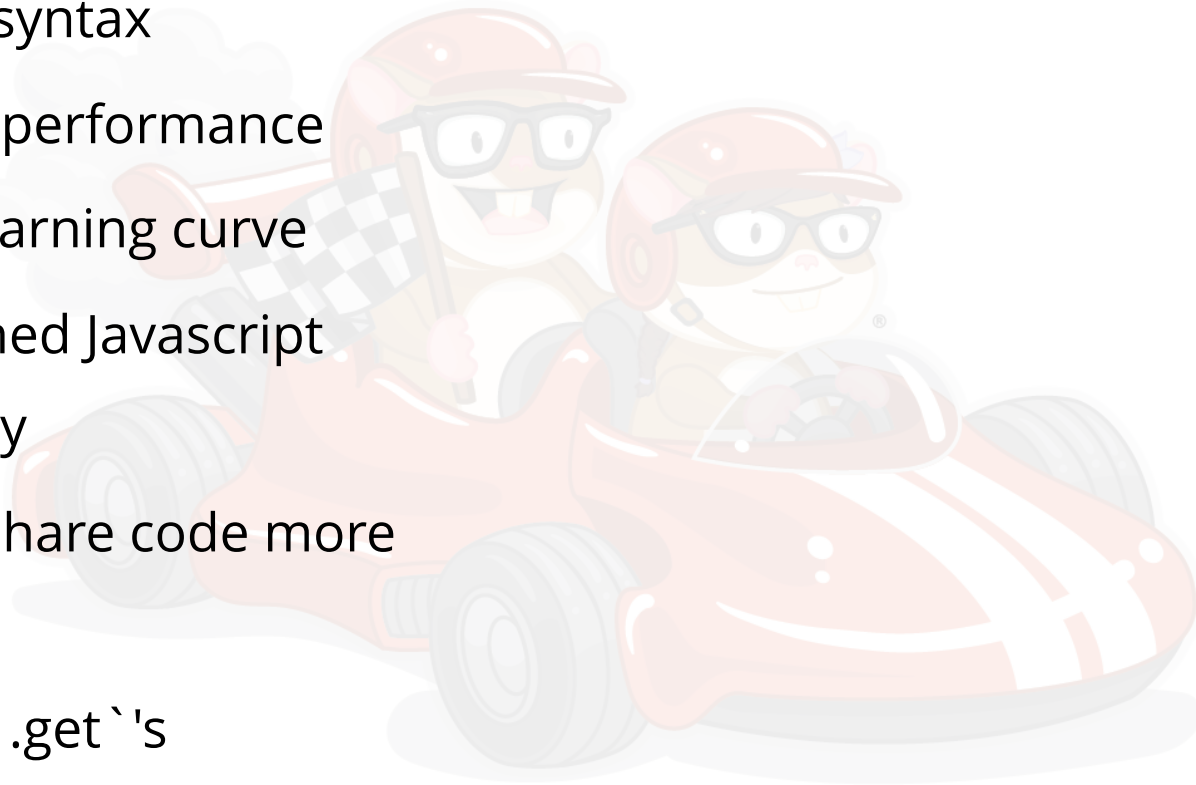
Native Classes

- ES6 Class syntax
- Increased performance
- Smooth learning curve
- More aligned Javascript community
- Ability to share code more easily
- No more ``.get``'s



Native Classes

- ES6 Class syntax
- Increased performance
- Smooth learning curve
- More aligned Javascript community
- Ability to share code more easily
- No more ``.get``'s
- Cleaner and easier to read



Native Classes

- ES6 Class syntax
- Increased performance
- Smooth learning curve
- More aligned Javascript community
- Ability to share code more easily
- No more ``.get``'s
- Cleaner and easier to read

// Classic Ember Object Syntax

```
1 import EmberObject, { computed } from '@ember/object';
2
3 const Person = EmberObject.extend({
4   init(props) {
5     this._super(props);
6     console.log(`Created ${this.get('fullName')}...`);
7   },
8
9   fullName: computed('firstName', 'lastName', function() {
10    return `${this.get('firstName')} ${this.get('lastName')}`;
11  })
12 });
13
14 let phoenix = Person.create({ firstName: 'Jean', lastName: 'Gray' });
```

// Native Class Syntax

```
1 import { computed } from '@ember/object'
2
3 class Person {
4   constructor(firstName, lastName) {
5     this.firstName = firstName;
6     this.lastName = lastName;
7     console.log(`Created ${this.fullName}...`);
8   }
9
10  @computed('firstName', 'lastName')
11  get fullName() {
12    return `${this.firstName} ${this.lastName}`;
13  }
14 }
15
16 let phoenix = new Person('Jean', 'Gray');
```

Native Classes

- ES6 Class syntax
- Increased performance
- Smooth learning curve
- More aligned Javascript community
- Ability to share code more easily
- No more `.get`'s
- Cleaner and easier to read

// Classic Ember Object Syntax

```
1 import EmberObject, { computed } from '@ember/object';
2
3 const Person = EmberObject.extend({
4   init(props) {
5     this._super(props);
6     console.log(`Created ${this.get('fullName')}...`);
7   },
8
9   fullName: computed('firstName', 'lastName', function() {
10    return `${this.get('firstName')} ${this.get('lastName')}`;
11  })
12 });
13
14 let phoenix = Person.create({ firstName: 'Jean', lastName: 'Gray' });
```

// Native Class Syntax

```
1 import { computed } from '@ember/object'
2
3 class Person {
4   constructor(firstName, lastName) {
5     this.firstName = firstName;
6     this.lastName = lastName;
7     console.log(`Created ${this.fullName}...`);
8   }
9
10  @computed('firstName', 'lastName')
11  get fullName() {
12    return `${this.firstName} ${this.lastName}`;
13  }
14 }
15
16 let phoenix = new Person('Jean', 'Gray');
```

Native Classes

- ES6 Class syntax
- Increased performance
- Smooth learning curve
- More aligned Javascript community
- Ability to share code more easily
- No more ``.get``'s
- Cleaner and easier to read

// Classic Ember Object Syntax

```
1 import EmberObject, { computed } from '@ember/object';
2
3 const Person = EmberObject.extend({
4   init(props) {
5     this._super(props);
6     console.log(`Created ${this.get('fullName')}...`);
7   },
8
9   fullName: computed('firstName', 'lastName', function() {
10    return `${this.get('firstName')} ${this.get('lastName')}`;
11  })
12 });
13
14 let phoenix = Person.create({ firstName: 'Jean', lastName: 'Gray' });
```

// Native Class Syntax

```
1 import { computed } from '@ember/object'
2
3 class Person {
4   constructor(firstName, lastName) {
5     this.firstName = firstName;
6     this.lastName = lastName;
7     console.log(`Created ${this.fullName}...`);
8   }
9
10  @computed('firstName', 'lastName')
11  get fullName() {
12    return `${this.firstName} ${this.lastName}`;
13  }
14 }
15
16 let phoenix = new Person('Jean', 'Gray');
```

Native Classes

- ES6 Class syntax
- Increased performance
- Smooth learning curve
- More aligned Javascript community
- Ability to share code more easily
- No more ``.get``'s
- Cleaner and easier to read

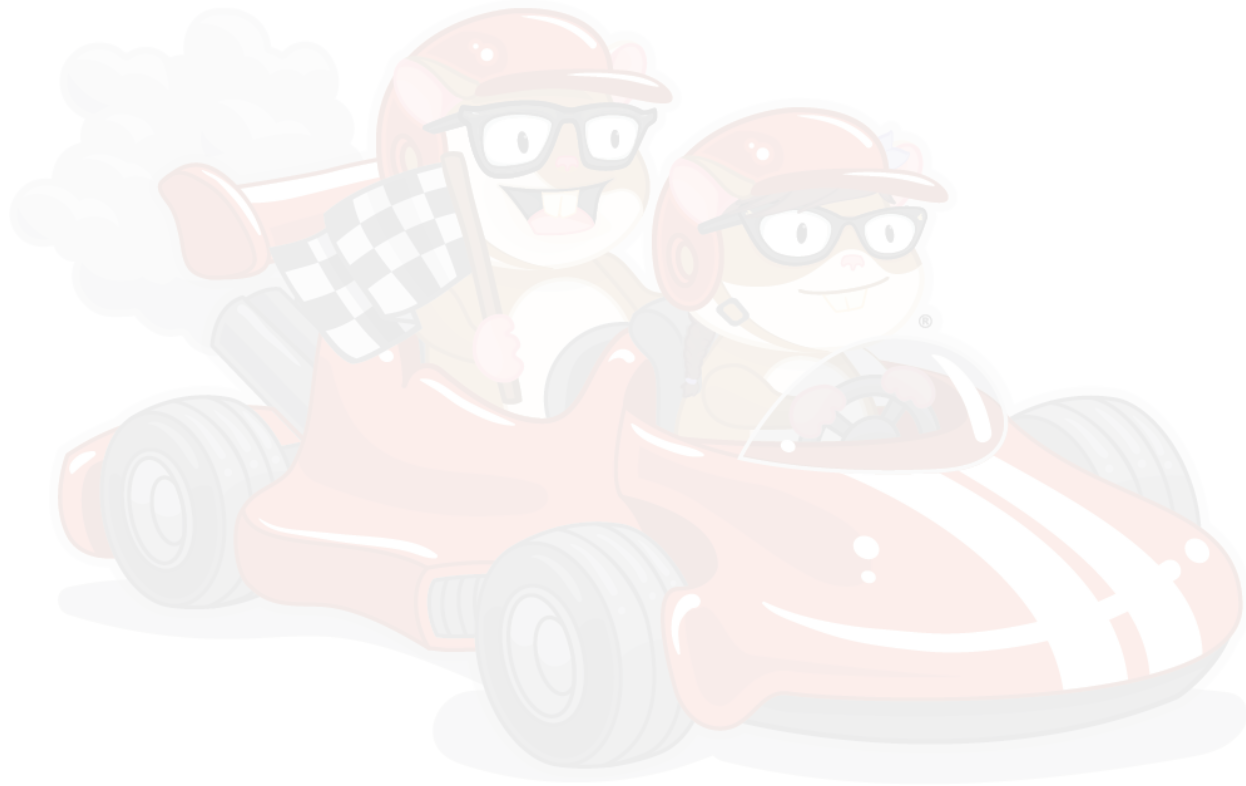
// Classic Ember Object Syntax

```
1 import EmberObject, { computed } from '@ember/object';
2
3 const Person = EmberObject.extend({
4   init(props) {
5     this._super(props);
6     console.log(`Created ${this.get('fullName')}...`);
7   },
8
9   fullName: computed('firstName', 'lastName', function() {
10    return `${this.get('firstName')} ${this.get('lastName')}`;
11  })
12 });
13
14 let phoenix = Person.create({ firstName: 'Jean', lastName: 'Gray' });
```

// Native Class Syntax

```
1 import { computed } from '@ember/object'
2
3 class Person {
4   constructor(firstName, lastName) {
5     this.firstName = firstName;
6     this.lastName = lastName;
7     console.log(`Created ${this.fullName}...`);
8   }
9
10  @computed('firstName', 'lastName')
11  get fullName() {
12    return `${this.firstName} ${this.lastName}`;
13  }
14 }
15
16 let phoenix = new Person('Jean', 'Gray');
```


Glimmer Components



Glimmer Components

- Offers simpler, ergonomic, and declarative approach

```
// Classic Component
```

```
1 import Component from '@ember/component';
2
3 export default Component.extend({
4   tagName: 'label',
5   noOfGuests: 5,
6   isMaxGuestExceeded: computed('noOfGuests') {
7     return this.noOfGuests > this.maxGuests;
8   }
9 });
```

```
// Glimmer Component
```

```
1 import Component from "@glimmer/component";
2 import { tracked } from "@glimmer/tracking";
3
4 export default class AddRsvp extends Component {
5   @tracked noOfGuests = 5;
6   get isMaxGuestExceeded() {
7     return this.noOfGuests > this.args.maxGuests;
8   }
9 }
10
11
12 }
```

Glimmer Components

- Offers simpler, ergonomic, and declarative approach
- Easier to understand

```
// Classic Component
```

```
1 import Component from '@ember/component';
2
3 export default Component.extend({
4   tagName: 'label',
5   noOfGuests: 5,
6   isMaxGuestExceeded: computed('noOfGuests') {
7     return this.noOfGuests > this.maxGuests;
8   }
9 });
```

```
// Glimmer Component
```

```
1 import Component from "@glimmer/component";
2 import { tracked } from "@glimmer/tracking";
3
4 export default class AddRsvp extends Component {
5   @tracked noOfGuests = 5;
6   get isMaxGuestExceeded() {
7     return this.noOfGuests > this.args.maxGuests;
8   }
9 }
10
11
12 }
```

Glimmer Components

- Offers simpler, ergonomic, and declarative approach
- Easier to understand
- Fewer hooks and properties

// Classic Component

```
1 import Component from '@ember/component';
2
3 export default Component.extend({
4   tagName: 'label',
5   noOfGuests: 5,
6   isMaxGuestExceeded: computed('noOfGuests') {
7     return this.noOfGuests > this.maxGuests;
8   }
9 });
```

// Glimmer Component

```
1 import Component from "@glimmer/component";
2 import { tracked } from "@glimmer/tracking";
3
4 export default class AddRsvp extends Component {
5   @tracked noOfGuests = 5;
6   get isMaxGuestExceeded() {
7     return this.noOfGuests > this.args.maxGuests;
8   }
9 }
10
11
12 }
```

Glimmer Components

- Offers simpler, ergonomic, and declarative approach
- Easier to understand
- Fewer hooks and properties
- No more implicit wrappers

// Classic Component

```
1 import Component from '@ember/component';
2
3 export default Component.extend({
4   tagName: 'label',
5   noOfGuests: 5,
6   isMaxGuestExceeded: computed('noOfGuests') {
7     return this.noOfGuests > this.maxGuests;
8   }
9 });
```

// Glimmer Component

```
1 import Component from "@glimmer/component";
2 import { tracked } from "@glimmer/tracking";
3
4 export default class AddRsvp extends Component {
5   @tracked noOfGuests = 5;
6   get isMaxGuestExceeded() {
7     return this.noOfGuests > this.args.maxGuests;
8   }
9 }
10
11
12 }
```

Glimmer Components

- Offers simpler, ergonomic, and declarative approach
- Easier to understand
- Fewer hooks and properties
- No more implicit wrappers

// Classic Component

```
1 import Component from '@ember/component';
2
3 export default Component.extend({
4   tagName: 'label',
5   noOfGuests: 5,
6   isMaxGuestExceeded: computed('noOfGuests') {
7     return this.noOfGuests > this.maxGuests;
8   }
9 });
```

```
1 {{noOfGuests}}
```

// Glimmer Component

```
1 import Component from "@glimmer/component";
2 import { tracked } from "@glimmer/tracking";
3
4 export default class AddRsvp extends Component {
5   @tracked noOfGuests = 5;
6   get isMaxGuestExceeded() {
7     return this.noOfGuests > this.args.maxGuests;
8   }
9 }
10
11
12 }
```

Glimmer Components

- Offers simpler, ergonomic, and declarative approach
- Easier to understand
- Fewer hooks and properties
- No more implicit wrappers

// Classic Component

```
1 import Component from '@ember/component';
2
3 export default Component.extend({
4   tagName: 'label',
5   noOfGuests: 5,
6   isMaxGuestExceeded: computed('noOfGuests') {
7     return this.noOfGuests > this.maxGuests;
8   }
9 });
```

```
1 {{noOfGuests}}
```

// Glimmer Component

```
1 import Component from "@glimmer/component";
2 import { tracked } from "@glimmer/tracking";
3
4 export default class AddRsvp extends Component {
5   @tracked noOfGuests = 5;
6   get isMaxGuestExceeded() {
7     return this.noOfGuests > this.args.maxGuests;
8   }
9 }
10
11
12 }
```

```
1 <label> {{this.noOfGuests}} </label>
```

Glimmer Components

- Offers simpler, ergonomic, and declarative approach
- Easier to understand
- Fewer hooks and properties
- No more implicit wrappers
- Namespaced arguments

// Classic Component

```
1 import Component from '@ember/component';
2
3 export default Component.extend({
4   tagName: 'label',
5   noOfGuests: 5,
6   isMaxGuestExceeded: computed('noOfGuests') {
7     return this.noOfGuests > this.maxGuests;
8   }
9 });
```

```
1 {{noOfGuests}}
```

// Glimmer Component

```
1 import Component from "@glimmer/component";
2 import { tracked } from "@glimmer/tracking";
3
4 export default class AddRsvp extends Component {
5   @tracked noOfGuests = 5;
6
7   get isMaxGuestExceeded() {
8     return this.noOfGuests > this.args.maxGuests;
9   }
10 }
11
12 }
```

```
1 <label> {{this.noOfGuests}} </label>
```


Glimmer Components

- Offers simpler, ergonomic, and declarative approach
- Easier to understand
- Fewer hooks and properties
- No more implicit wrappers
- Namespaced arguments

// Classic Component

```
1 import Component from '@ember/component';
2
3 export default Component.extend({
4   tagName: 'label',
5   noOfGuests: 5,
6   isMaxGuestExceeded: computed('noOfGuests') {
7     return this.noOfGuests > this.maxGuests;
8   }
9 });
```

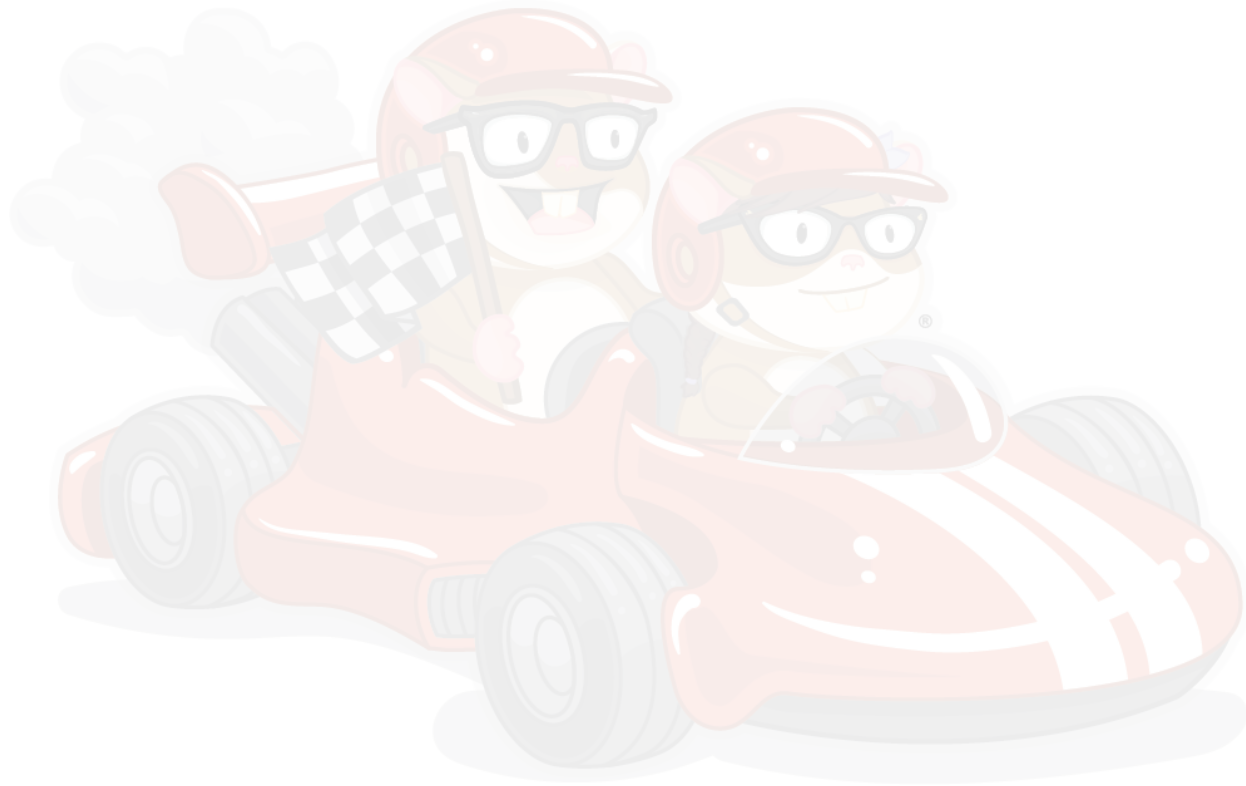
```
1 {{noOfGuests}}
```

// Glimmer Component

```
1 import Component from "@glimmer/component";
2 import { tracked } from "@glimmer/tracking";
3
4 export default class AddRsvp extends Component {
5   @tracked noOfGuests = 5;
6
7   get isMaxGuestExceeded() {
8     return this.noOfGuests > this.args.maxGuests;
9   }
10 }
11
12 }
```

```
1 <label> {{this.noOfGuests}} </label>
```

Templating in Octane



Templating in Octane

- Angle Brackets syntax
 - In line with the HTML standards
 - Capital Case Syntax
 - Easy to distinguish from helpers, properties etc.

```
// Classic templating syntax
```

```
1 {{employee-details
2   name=employeeName
3   empId=employeeId
4   addEmployee=(action 'addEmployee')
5 }}
6
```

```
// Octane templating syntax
```

```
1 <EmployeeDetails
2   @name={{this.employeeName}}
3   @empId={{@employeeId}}
4   @addEmployee={{this.addEmployee}}
5 />
```

Templating in Octane

- Angle Brackets syntax
 - In line with the HTML standards
 - Capital Case Syntax
 - Easy to distinguish from helpers, properties etc.
- Named arguments
 - Denoted with `@` symbol
 - Easy differentiation from local and external properties

```
// Classic templating syntax
```

```
1 {{employee-details
2   name=employeeName
3   empId=employeeId
4   addEmployee=(action 'addEmployee')
5 }}
6
```

```
// Octane templating syntax
```

```
1 <EmployeeDetails
2   @name={{this.employeeName}}
3   @empId={{@employeeId}}
4   @addEmployee={{this.addEmployee}}
5 />
```

Templating in Octane

- Angle Brackets syntax
 - In line with the HTML standards
 - Capital Case Syntax
 - Easy to distinguish from helpers, properties etc.
- Named arguments
 - Denoted with `@` symbol
 - Easy differentiation from local and external properties
- Required `this`
 - Provides clear point-of-origin
 - Easy to read and understand

```
// Classic templating syntax
```

```
1 {{employee-details
2   name=employeeName
3   empId=employeeId
4   addEmployee=(action 'addEmployee')
5 }}
6
```

```
// Octane templating syntax
```

```
1 <EmployeeDetails
2   @name={{this.employeeName}}
3   @empId={{@employeeId}}
4   @addEmployee={{this.addEmployee}}
5 />
```

Templating in Octane

- Angle Brackets syntax
 - In line with the HTML standards
 - Capital Case Syntax
 - Easy to distinguish from helpers, properties etc.
- Named arguments
 - Denoted with `@` symbol
 - Easy differentiation from local and external properties
- Required `this`
 - Provides clear point-of-origin
 - Easy to read and understand

// Classic templating syntax

```
1 {{employee-details
2   name=employeeName
3   empId=employeeId
4   addEmployee=(action 'addEmployee')
5 }}
6
```

// Octane templating syntax

```
1 <EmployeeDetails
2   @name={{this.employeeName}}
3   @empId={{@employeeId}}
4   @addEmployee={{this.addEmployee}}
5 />
```

Tracked Properties

- @tracked syntax
- Explicit declarations
- Cleaner code
- Reduces complexity
- No More ``.set``'s

// Classic syntax

```
1 import EmberObject, { computed } from '@ember/object';
2
3 const Person = EmberObject.extend({
4   firstName: 'Tom',
5   lastName: 'Dale',
6   count: 0,
7
8   fullName: computed('firstName', 'lastName', function() {
9     this.set('count', this.get('count') + 1);
10    return `${this.firstName} ${this.lastName}`;
11  }),
12 });
```

// @tracked syntax

```
1 import { tracked } from '@glimmer/tracking';
2
3 class Person {
4   @tracked firstName = 'Tom';
5   @tracked lastName = 'Dale';
6   @tracked count = 0;
7
8   get fullName() {
9     this.count = this.count + 1;
10    return `${this.firstName} ${this.lastName}`;
11  }
12 }
```

Tracked Properties

- @tracked syntax
- Explicit declarations
- Cleaner code
- Reduces complexity
- No More `.set`'s`

// Classic syntax

```
1 import EmberObject, { computed } from '@ember/object';
2
3 const Person = EmberObject.extend({
4   firstName: 'Tom',
5   lastName: 'Dale',
6   count: 0,
7
8   fullName: computed('firstName', 'lastName', function() {
9     this.set('count', this.get('count') + 1);
10    return `${this.firstName} ${this.lastName}`;
11  }),
12 });
```

// @tracked syntax

```
1 import { tracked } from '@glimmer/tracking';
2
3 class Person {
4   @tracked firstName = 'Tom';
5   @tracked lastName = 'Dale';
6   @tracked count = 0;
7
8   get fullName() {
9     this.count = this.count + 1;
10    return `${this.firstName} ${this.lastName}`;
11  }
12 }
```


Tracked Properties

- @tracked syntax
- Explicit declarations
- Cleaner code
- Reduces complexity
- No More ``.set``'s

// Classic syntax

```
1 import EmberObject, { computed } from '@ember/object';
2
3 const Person = EmberObject.extend({
4   firstName: 'Tom',
5   lastName: 'Dale',
6   count: 0,
7
8   fullName: computed('firstName', 'lastName', function() {
9     this.set('count', this.get('count') + 1);
10    return `${this.firstName} ${this.lastName}`;
11  }),
12 });
```

// @tracked syntax

```
1 import { tracked } from '@glimmer/tracking';
2
3 class Person {
4   @tracked firstName = 'Tom';
5   @tracked lastName = 'Dale';
6   @tracked count = 0;
7
8   get fullName() {
9     this.count = this.count + 1;
10    return `${this.firstName} ${this.lastName}`;
11  }
12 }
```

Tracked Properties

- @tracked syntax
- Explicit declarations
- Cleaner code
- Reduces complexity
- No More ``.set``'s

// Classic syntax

```
1 import EmberObject, { computed } from '@ember/object';
2
3 const Person = EmberObject.extend({
4   firstName: 'Tom',
5   lastName: 'Dale',
6   count: 0,
7
8   fullName: computed('firstName', 'lastName', function() {
9     this.set('count', this.get('count') + 1);
10    return `${this.firstName} ${this.lastName}`;
11  }),
12 });
```

// @tracked syntax

```
1 import { tracked } from '@glimmer/tracking';
2
3 class Person {
4   @tracked firstName = 'Tom';
5   @tracked lastName = 'Dale';
6   @tracked count = 0;
7
8   get fullName() {
9     this.count = this.count + 1;
10    return `${this.firstName} ${this.lastName}`;
11  }
12 }
```

Tracked Properties

- @tracked syntax
- Explicit declarations
- Cleaner code
- Reduces complexity
- No More ``.set``'s

// Classic syntax

```
1 import EmberObject, { computed } from '@ember/object';
2
3 const Person = EmberObject.extend({
4   firstName: 'Tom',
5   lastName: 'Dale',
6   count: 0,
7
8   fullName: computed('firstName', 'lastName', function() {
9     this.set('count', this.get('count') + 1);
10    return `${this.firstName} ${this.lastName}`;
11  }),
12 });
```

// @tracked syntax

```
1 import { tracked } from '@glimmer/tracking';
2
3 class Person {
4   @tracked firstName = 'Tom';
5   @tracked lastName = 'Dale';
6   @tracked count = 0;
7
8   get fullName() {
9     this.count = this.count + 1;
10    return `${this.firstName} ${this.lastName}`;
11  }
12 }
```

Tracked Properties

- @tracked syntax
- Explicit declarations
- Cleaner code
- Reduces complexity
- No More ``.set``'s

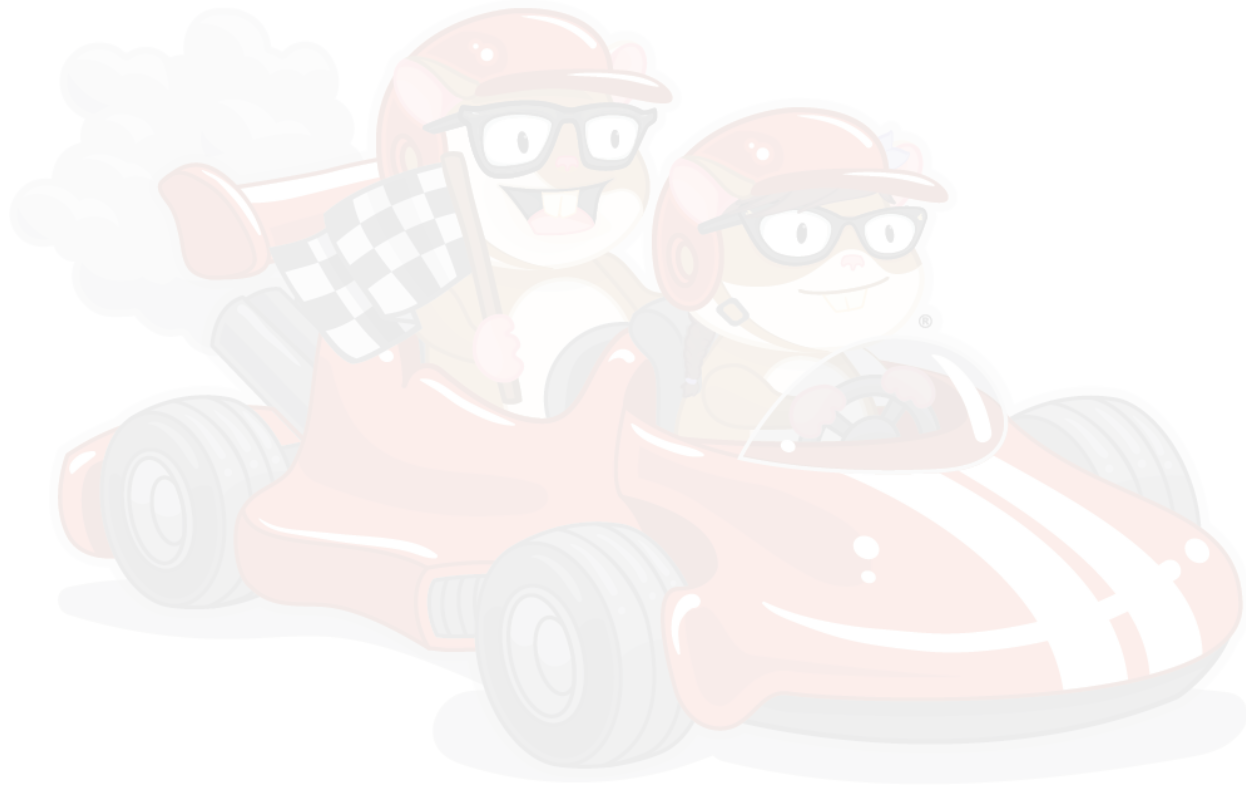
// Classic syntax

```
1 import EmberObject, { computed } from '@ember/object';
2
3 const Person = EmberObject.extend({
4   firstName: 'Tom',
5   lastName: 'Dale',
6   count: 0,
7
8   fullName: computed('firstName', 'lastName', function() {
9     this.set('count', this.get('count') + 1);
10    return `${this.firstName} ${this.lastName}`;
11  }),
12 });
```

// @tracked syntax

```
1 import { tracked } from '@glimmer/tracking';
2
3 class Person {
4   @tracked firstName = 'Tom';
5   @tracked lastName = 'Dale';
6   @tracked count = 0;
7
8   get fullName() {
9     this.count = this.count + 1;
10    return `${this.firstName} ${this.lastName}`;
11  }
12 }
```

Modifiers and Decorators



Modifiers and Decorators

- Modifiers:

- Functions or classes used directly in templates
- Applied directly to elements
- Allows targeting specific elements more easily
- Easy to reuse

```
1 <span>{{this.formattedCount}}</span>
2
3 <button {{on "click" this.increment}}>Increment</button>
4
5 <button {{on "click" this.decrement}}>Decrement</button>
6
```

Modifiers and Decorators

- Modifiers:

- Functions or classes used directly in templates
- Applied directly to elements
- Allows targeting specific elements more easily
- Easy to reuse

```
1 <span>{{this.formattedCount}}</span>
2
3 <button {{on "click" this.increment}}>Increment</button>
4
5 <button {{on "click" this.decrement}}>Decrement</button>
6
```

- Decorators:

- Abstracts functionality
- Improved Developer experience

```
1 import Component from '@glimmer/component';
2 import { tracked } from '@glimmer/tracking';
3 import { action } from '@ember/object';
4
5 export default class Counter extends Component {
6   @tracked count = 0;
7
8   get formattedCount() {
9     return this.count.toString().padStart(3, '0');
10  }
11
12   @action
13   increment() {
14     this.count++;
15   }
16
17   @action
18   decrement() {
19     this.count--;
20   }
21 }
```

Modifiers and Decorators

- Modifiers:

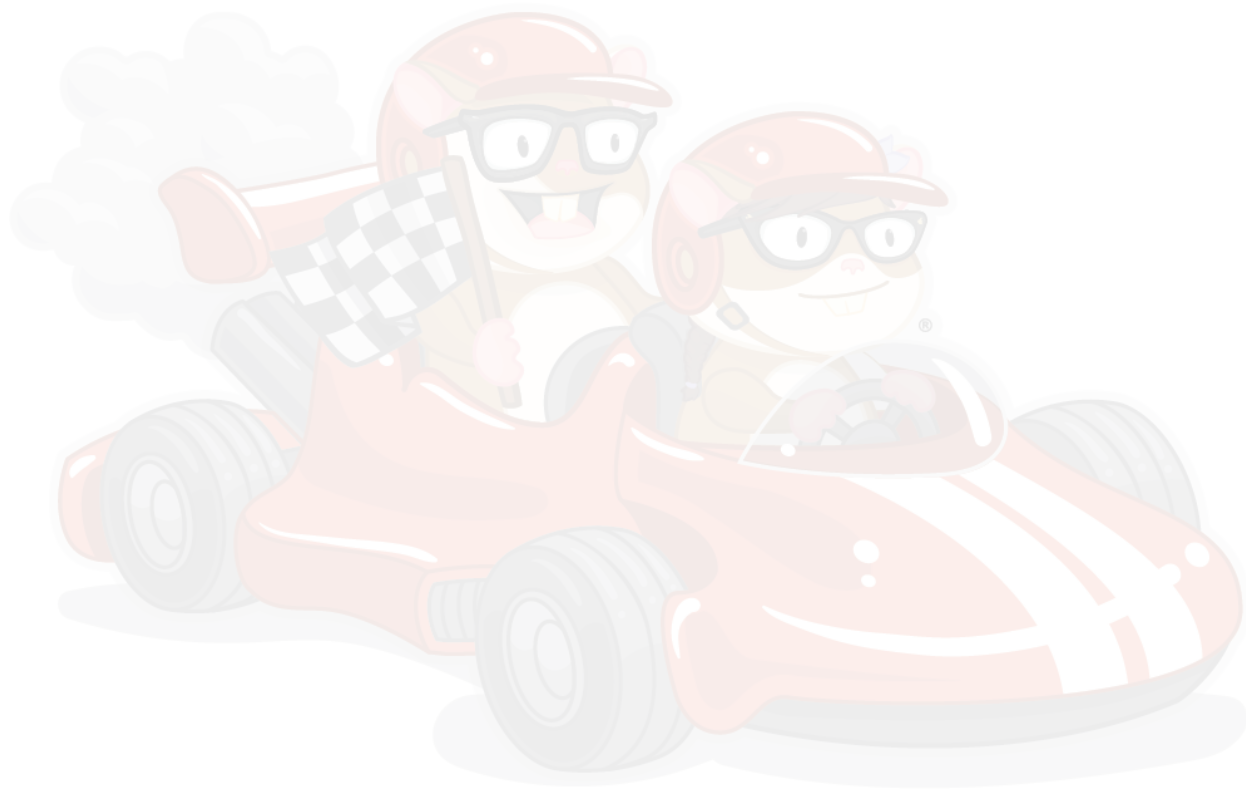
- Functions or classes used directly in templates
- Applied directly to elements
- Allows targeting specific elements more easily
- Easy to reuse

```
1 <span>{{this.formattedCount}}</span>
2
3 <button {{on "click" this.increment}}>Increment</button>
4
5 <button {{on "click" this.decrement}}>Decrement</button>
6
```

- Decorators:

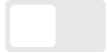
- Abstracts functionality
- Improved Developer experience

```
1 import Component from '@glimmer/component';
2 import { tracked } from '@glimmer/tracking';
3 import { action } from '@ember/object';
4
5 export default class Counter extends Component {
6   @tracked count = 0;
7
8   get formattedCount() {
9     return this.count.toString().padStart(3, '0');
10  }
11
12   @action
13   increment() {
14     this.count++;
15   }
16
17   @action
18   decrement() {
19     this.count--;
20   }
21 }
```

```
1 {{x-toggle-label
2   label=labelVal
3   value=false
4   sendToggle=(action 'sendToggle')
5 }}
```

customOffLabel



// Classic Component syntax

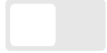
```
1 import { readOnly } from '@ember/object/computed';
2 import Component from '@ember/component';
3 import { computed } from '@ember/object';
4 import layout from './template';
5
6 export default Component.extend({
7   layout,
8   tagName: 'label',
9   attributeBindings: ['for'],
10  classNames: ['toggle-text', 'toggle-prefix'],
11  classNameBindings: ['labelType'],
12  for: readOnly('switchId'),
13  isVisible: readOnly('show'),
14  isAwesome: true,
15
16  labelValue: computed('isAwesome', function() {
17    return this.get('isAwesome') ?
18      'awesome-label' : 'lesser-awesome-label';
19  }),
20
21  type: computed('value', {
22    get() {
23      return this.get('value') ? 'on' : 'off';
24    }
25  }),
26
27  click(e) {
28    e.stopPropagation();
29    e.preventDefault();
30    this.sendToggle(this.get('value'));
31  }
32 });
```



```
1 {{#if hasBlock}}
2   {{yield label}}
3 {{else}}
4   {{label}}
5 {{/if}}
6
```



customOffLabel



```

1  {{x-toggle-label
2    label=labelVal
3    value=false
4    sendToggle=(action 'sendToggle')
5  }}

```

// Classic Component syntax

```

1  import { readOnly } from '@ember/object/computed';
2  import Component from '@ember/component';
3  import { computed } from '@ember/object';
4  import layout from './template';
5
6  export default Component.extend({
7    layout,
8    tagName: 'label',
9    attributeBindings: ['for'],
10   classNames: ['toggle-text', 'toggle-prefix'],
11   classNameBindings: ['labelType'],
12   for: readOnly('switchId'),
13   isVisible: readOnly('show'),
14   isAwesome: true,
15
16   labelValue: computed('isAwesome', function() {
17     return this.get('isAwesome') ?
18       'awesome-label' : 'lesser-awesome-label';
19   }),
20
21   type: computed('value', {
22     get() {
23       return this.get('value') ? 'on' : 'off';
24     }
25   }),
26
27   click(e) {
28     e.stopPropagation();
29     e.preventDefault();
30     this.sendToggle(this.get('value'));
31   }
32 });

```



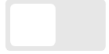
```

1  {{#if hasBlock}}
2    {{yield label}}
3  {{else}}
4    {{label}}
5  {{/if}}
6

```



customOffLabel



```

1  {{x-toggle-label
2     label=labelVal
3     value=false
4     sendToggle=(action 'sendToggle')
5  }}

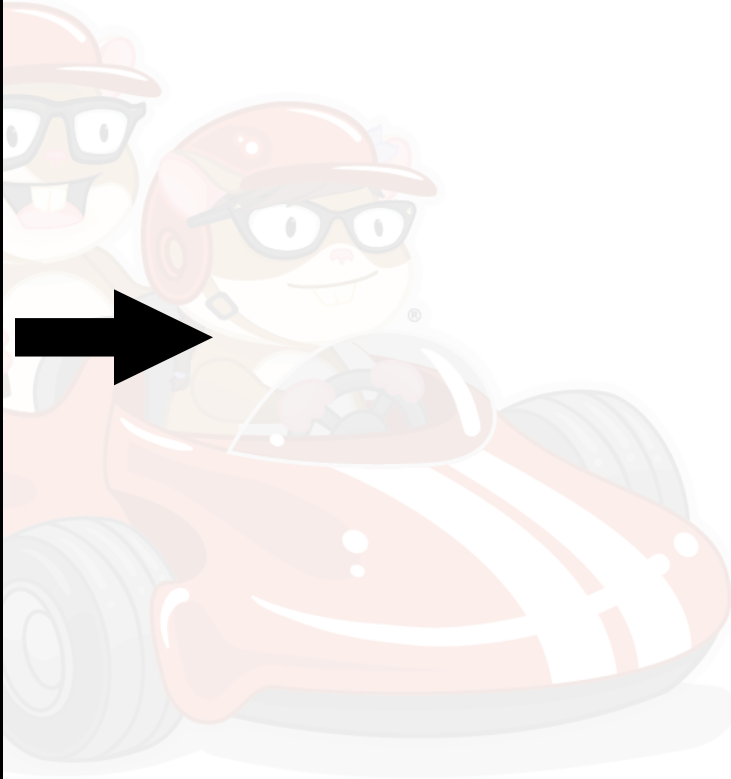
```

// Classic Component syntax

```

1  import { readOnly } from '@ember/object/computed';
2  import Component from '@ember/component';
3  import { computed } from '@ember/object';
4  import layout from './template';
5
6  export default Component.extend({
7     layout,
8     tagName: 'label',
9     attributeBindings: ['for'],
10    classNames: ['toggle-text', 'toggle-prefix'],
11    classNameBindings: ['labelType'],
12    for: readOnly('switchId'),
13    isVisible: readOnly('show'),
14    isAwesome: true,
15
16    labelValue: computed('isAwesome', function() {
17      return this.get('isAwesome') ?
18        'awesome-label' : 'lesser-awesome-label';
19    }),
20
21    type: computed('value', {
22      get() {
23        return this.get('value') ? 'on' : 'off';
24      }
25    }),
26
27    click(e) {
28      e.stopPropagation();
29      e.preventDefault();
30      this.sendToggle(this.get('value'));
31    }
32  });

```



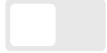
```

1  {{#if hasBlock}}
2    {{yield label}}
3  {{else}}
4    {{label}}
5  {{/if}}
6

```



customOffLabel



```

1  {{x-toggle-label
2    label=labelVal
3    value=false
4    sendToggle=(action 'sendToggle')
5  }}

```

// Classic Component syntax

```

1  import { readOnly } from '@ember/object/computed';
2  import Component from '@ember/component';
3  import { computed } from '@ember/object';
4  import layout from './template';
5
6  export default Component.extend({
7    layout,
8    tagName: 'label',
9    attributeBindings: ['for'],
10   classNames: ['toggle-text', 'toggle-prefix'],
11   classNameBindings: ['labelType'],
12   for: readOnly('switchId'),
13   isVisible: readOnly('show'),
14   isAwesome: true,
15
16   labelValue: computed('isAwesome', function() {
17     return this.get('isAwesome') ?
18       'awesome-label' : 'lesser-awesome-label';
19   }),
20
21   type: computed('value', {
22     get() {
23       return this.get('value') ? 'on' : 'off';
24     }
25   }),
26
27   click(e) {
28     e.stopPropagation();
29     e.preventDefault();
30     this.sendToggle(this.get('value'));
31   }
32 });

```

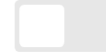
```

1  {{#if hasBlock}}
2    {{yield label}}
3  {{else}}
4    {{label}}
5  {{/if}}
6

```



customOffLabel



```

1  <ToggleLabel
2    @label={{this.labelVal}}
3    @value=false
4    sendToggle={{this.sendToggle}}
5  />

```

// Octane Component syntax

```

1  import Component from '@glimmer/component';
2  import { tracked } from '@glimmer/tracking';
3  import { action } from '@ember/object';
4
5  export default class ToggleLabel extends Component {
6    @tracked isAwesome = true;
7
8    get type() {
9      return this.args.value ? 'on' : 'off';
10   }
11
12   get labelValue() {
13     return this.isAwesome ?
14       'awesome-label' : 'lesser-awesome-label';
15   }
16
17   @action
18   handleClick(e) {
19     e.stopPropagation();
20     e.preventDefault();
21     this.args.sendToggle(this.args.value);
22   }
23 }
24

```

```

1  <label
2    for="{{@switchId}}"
3    {{on 'click' this.handleClick}}
4
5    class="
6      toggle-text
7      toggle-prefix
8      {{this.type}}-label
9      {{if @show 'is-visible' 'is-hidden'}}
10   "
11  >
12   {{#if @hasBlock}}
13     {{yield @label}}
14   {{else}}
15     {{@label}}
16   {{/if}}
17 </label>
18

```



```

1  {{x-toggle-label
2    label=labelVal
3    value=false
4    sendToggle=(action 'sendToggle')
5  }}

```

customOffLabel

// Classic Component syntax

```

1  import { readOnly } from '@ember/object/computed';
2  import Component from '@ember/component';
3  import { computed } from '@ember/object';
4  import layout from './template';
5
6  export default Component.extend({
7    layout,
8    tagName: 'label',
9    attributeBindings: ['for'],
10   classNames: ['toggle-text', 'toggle-prefix'],
11   classNameBindings: ['labelType'],
12   for: readOnly('switchId'),
13   isVisible: readOnly('show'),
14   isAwesome: true,
15
16   labelValue: computed('isAwesome', function() {
17     return this.get('isAwesome') ?
18       'awesome-label' : 'lesser-awesome-label';
19   }),
20
21   type: computed('value', {
22     get() {
23       return this.get('value') ? 'on' : 'off';
24     }
25   }),
26
27   click(e) {
28     e.stopPropagation();
29     e.preventDefault();
30     this.sendToggle(this.get('value'));
31   }
32 });

```

```

1  {{#if hasBlock}}
2    {{yield label}}
3  {{else}}
4    {{label}}
5  {{/if}}
6

```



```

1  <ToggleLabel
2    @label={{this.labelVal}}
3    @value=false
4    sendToggle={{this.sendToggle}}
5  />

```

customOffLabel

// Octane Component syntax

```

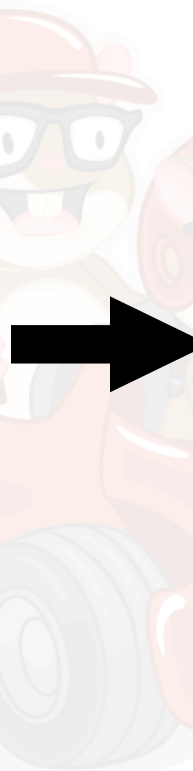
1  import Component from '@glimmer/component';
2  import { tracked } from '@glimmer/tracking';
3  import { action } from '@ember/object';
4
5  export default class ToggleLabel extends Component {
6    @tracked isAwesome = true;
7
8    get type() {
9      return this.args.value ? 'on' : 'off';
10   }
11
12   get labelValue() {
13     return this.isAwesome ?
14       'awesome-label' : 'lesser-awesome-label';
15   }
16
17   @action
18   handleClick(e) {
19     e.stopPropagation();
20     e.preventDefault();
21     this.args.sendToggle(this.args.value);
22   }
23 }
24

```

```

1  <label
2    for="{{@switchId}}"
3    {{on 'click' this.handleClick}}
4
5    class="
6      toggle-text
7      toggle-prefix
8      {{this.type}}-label
9      {{if @show 'is-visible' 'is-hidden'}}
10   "
11  >
12   {{#if @hasBlock}}
13     {{yield @label}}
14   {{else}}
15     {{@label}}
16   {{/if}}
17 </label>
18

```





```

1 {{x-toggle-label
2   label=labelVal
3   value=false
4   sendToggle=(action 'sendToggle')
5 }}

```

customOffLabel

// Classic Component syntax

```

1 import { readOnly } from '@ember/object/computed';
2 import Component from '@ember/component';
3 import { computed } from '@ember/object';
4 import layout from './template';
5
6 export default Component.extend({
7   layout,
8   tagName: 'label',
9   attributeBindings: ['for'],
10  classNames: ['toggle-text', 'toggle-prefix'],
11  classNameBindings: ['labelType'],
12  for: readOnly('switchId'),
13  isVisible: readOnly('show'),
14  isAwesome: true,
15
16  labelValue: computed('isAwesome', function() {
17    return this.get('isAwesome') ?
18      'awesome-label' : 'lesser-awesome-label';
19  }),
20
21  type: computed('value', {
22    get() {
23      return this.get('value') ? 'on' : 'off';
24    }
25  }),
26
27  click(e) {
28    e.stopPropagation();
29    e.preventDefault();
30    this.sendToggle(this.get('value'));
31  }
32 });

```

```

1 {{#if hasBlock}}
2   {{yield label}}
3 {{else}}
4   {{label}}
5 {{/if}}
6

```



```

1 <ToggleLabel
2   @label={{this.labelVal}}
3   @value=false
4   sendToggle={{this.sendToggle}}
5 />

```

customOffLabel

// Octane Component syntax

```

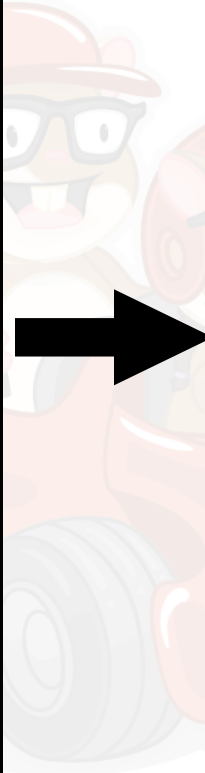
1 import Component from '@glimmer/component';
2 import { tracked } from '@glimmer/tracking';
3 import { action } from '@ember/object';
4
5 export default class ToggleLabel extends Component {
6   @tracked isAwesome = true;
7
8   get type() {
9     return this.args.value ? 'on' : 'off';
10  }
11
12  get labelValue() {
13    return this.isAwesome ?
14      'awesome-label' : 'lesser-awesome-label';
15  }
16
17  @action
18  handleClick(e) {
19    e.stopPropagation();
20    e.preventDefault();
21    this.args.sendToggle(this.args.value);
22  }
23 }
24

```

```

1 <label
2   for="{{@switchId}}"
3   {{on 'click' this.handleClick}}
4
5   class="
6     toggle-text
7     toggle-prefix
8     {{this.type}}-label
9     {{if @show 'is-visible' 'is-hidden'}}
10  "
11 >
12   {{#if @hasBlock}}
13     {{yield @label}}
14   {{else}}
15     {{@label}}
16   {{/if}}
17 </label>
18

```





```

1  {{x-toggle-label
2    label=labelVal
3    value=false
4    sendToggle=(action 'sendToggle')
5  }}

```

customOffLabel

// Classic Component syntax

```

1  import { readOnly } from '@ember/object/computed';
2  import Component from '@ember/component';
3  import { computed } from '@ember/object';
4  import layout from './template';
5
6  export default Component.extend({
7    layout,
8    tagName: 'label',
9    attributeBindings: ['for'],
10   classNames: ['toggle-text', 'toggle-prefix'],
11   classNameBindings: ['labelType'],
12   for: readOnly('switchId'),
13   isVisible: readOnly('show'),
14   isAwesome: true,
15
16   labelValue: computed('isAwesome', function() {
17     return this.get('isAwesome') ?
18       'awesome-label' : 'lesser-awesome-label';
19   }),
20
21   type: computed('value', {
22     get() {
23       return this.get('value') ? 'on' : 'off';
24     }
25   }),
26
27   click(e) {
28     e.stopPropagation();
29     e.preventDefault();
30     this.sendToggle(this.get('value'));
31   }
32 });

```

```

1  {{#if hasBlock}}
2    {{yield label}}
3  {{else}}
4    {{label}}
5  {{/if}}
6

```



```

1  <ToggleLabel
2    @label={{this.labelVal}}
3    @value=false
4    sendToggle={{this.sendToggle}}
5  />

```

customOffLabel

// Octane Component syntax

```

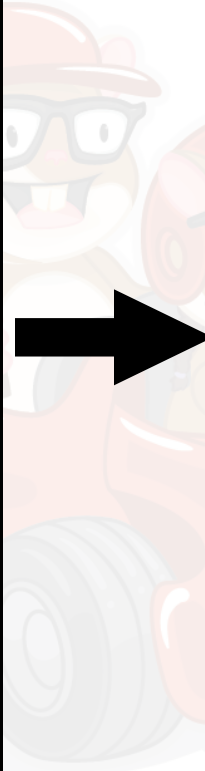
1  import Component from '@glimmer/component';
2  import { tracked } from '@glimmer/tracking';
3  import { action } from '@ember/object';
4
5  export default class ToggleLabel extends Component {
6    @tracked isAwesome = true;
7
8    get type() {
9      return this.args.value ? 'on' : 'off';
10   }
11
12   get labelValue() {
13     return this.isAwesome ?
14       'awesome-label' : 'lesser-awesome-label';
15   }
16
17   @action
18   handleClick(e) {
19     e.stopPropagation();
20     e.preventDefault();
21     this.args.sendToggle(this.args.value);
22   }
23 }
24

```

```

1  <label
2    for="{{@switchId}}"
3    {{on 'click' this.handleClick}}
4
5    class="
6      toggle-text
7      toggle-prefix
8      {{this.type}}-label
9      {{if @show 'is-visible' 'is-hidden'}}
10   "
11  >
12   {{#if @hasBlock}}
13     {{yield @label}}
14   {{else}}
15     {{@label}}
16   {{/if}}
17 </label>
18

```





```

1 {{x-toggle-label
2   label=labelVal
3   value=false
4   sendToggle=(action 'sendToggle')
5 }}

```

customOffLabel

// Classic Component syntax

```

1 import { readOnly } from '@ember/object/computed';
2 import Component from '@ember/component';
3 import { computed } from '@ember/object';
4 import layout from './template';
5
6 export default Component.extend({
7   layout,
8   tagName: 'label',
9   attributeBindings: ['for'],
10  classNames: ['toggle-text', 'toggle-prefix'],
11  classNameBindings: ['labelType'],
12  for: readOnly('switchId'),
13  isVisible: readOnly('show'),
14  isAwesome: true,
15
16  labelValue: computed('isAwesome', function() {
17    return this.get('isAwesome') ?
18      'awesome-label' : 'lesser-awesome-label';
19  }),
20
21  type: computed('value', {
22    get() {
23      return this.get('value') ? 'on' : 'off';
24    }
25  }),
26
27  click(e) {
28    e.stopPropagation();
29    e.preventDefault();
30    this.sendToggle(this.get('value'));
31  }
32 });

```

```

1 {{#if hasBlock}}
2   {{yield label}}
3 {{else}}
4   {{label}}
5 {{/if}}
6

```



```

1 <ToggleLabel
2   @label={{this.labelVal}}
3   @value=false
4   sendToggle={{this.sendToggle}}
5 />

```

customOffLabel

// Octane Component syntax

```

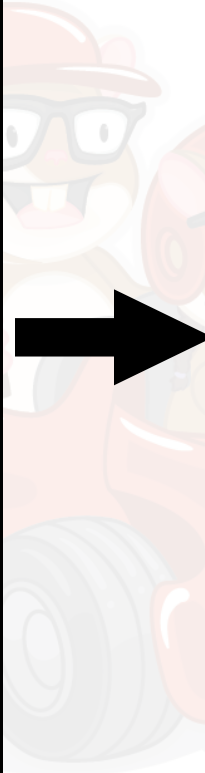
1 import Component from '@glimmer/component';
2 import { tracked } from '@glimmer/tracking';
3 import { action } from '@ember/object';
4
5 export default class ToggleLabel extends Component {
6   @tracked isAwesome = true;
7
8   get type() {
9     return this.args.value ? 'on' : 'off';
10  }
11
12  get labelValue() {
13    return this.isAwesome ?
14      'awesome-label' : 'lesser-awesome-label';
15  }
16
17  @action
18  handleClick(e) {
19    e.stopPropagation();
20    e.preventDefault();
21    this.args.sendToggle(this.args.value);
22  }
23 }
24

```

```

1 <label
2   for="{{@switchId}}"
3   {{on 'click' this.handleClick}}
4
5   class="
6     toggle-text
7     toggle-prefix
8     {{this.type}}-label
9     {{if @show 'is-visible' 'is-hidden'}}
10  "
11 >
12   {{#if @hasBlock}}
13     {{yield @label}}
14   {{else}}
15     {{@label}}
16   {{/if}}
17 </label>
18

```





```

1 {{x-toggle-label
2   label=labelVal
3   value=false
4   sendToggle=(action 'sendToggle')
5 }}

```

customOffLabel

// Classic Component syntax

```

1 import { readOnly } from '@ember/object/computed';
2 import Component from '@ember/component';
3 import { computed } from '@ember/object';
4 import layout from './template';
5
6 export default Component.extend({
7   layout,
8   tagName: 'label',
9   attributeBindings: ['for'],
10  classNames: ['toggle-text', 'toggle-prefix'],
11  classNameBindings: ['labelType'],
12  for: readOnly('switchId'),
13  isVisible: readOnly('show'),
14  isAwesome: true,
15
16  labelValue: computed('isAwesome', function() {
17    return this.get('isAwesome') ?
18      'awesome-label' : 'lesser-awesome-label';
19  }),
20
21  type: computed('value', {
22    get() {
23      return this.get('value') ? 'on' : 'off';
24    }
25  }),
26
27  click(e) {
28    e.stopPropagation();
29    e.preventDefault();
30    this.sendToggle(this.get('value'));
31  }
32 });

```

```

1 {{#if hasBlock}}
2   {{yield label}}
3 {{else}}
4   {{label}}
5 {{/if}}
6

```



```

1 <ToggleLabel
2   @label={{this.labelVal}}
3   @value=false
4   sendToggle={{this.sendToggle}}
5 />

```

customOffLabel

// Octane Component syntax

```

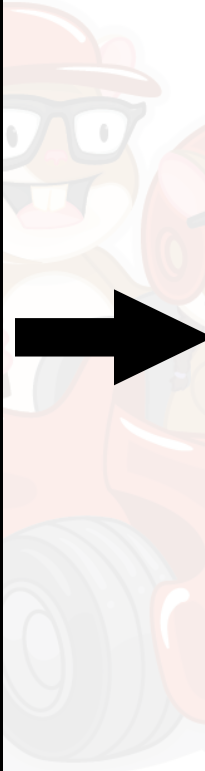
1 import Component from '@glimmer/component';
2 import { tracked } from '@glimmer/tracking';
3 import { action } from '@ember/object';
4
5 export default class ToggleLabel extends Component {
6   @tracked isAwesome = true;
7
8   get type() {
9     return this.args.value ? 'on' : 'off';
10  }
11
12  get labelValue() {
13    return this.isAwesome ?
14      'awesome-label' : 'lesser-awesome-label';
15  }
16
17  @action
18  handleClick(e) {
19    e.stopPropagation();
20    e.preventDefault();
21    this.args.sendToggle(this.args.value);
22  }
23 }
24

```

```

1 <label
2   for="{{@switchId}}"
3   {{on 'click' this.handleClick}}
4
5   class="
6     toggle-text
7     toggle-prefix
8     {{this.type}}-label
9     {{if @show 'is-visible' 'is-hidden'}}
10  "
11 >
12   {{#if @hasBlock}}
13     {{yield @label}}
14   {{else}}
15     {{@label}}
16   {{/if}}
17 </label>
18

```





```

1  {{x-toggle-label
2    label=labelVal
3    value=false
4    sendToggle=(action 'sendToggle')
5  }}

```

customOffLabel

// Classic Component syntax

```

1  import { readOnly } from '@ember/object/computed';
2  import Component from '@ember/component';
3  import { computed } from '@ember/object';
4  import layout from './template';
5
6  export default Component.extend({
7    layout,
8    tagName: 'label',
9    attributeBindings: ['for'],
10   classNameBindings: ['toggle-text', 'toggle-prefix'],
11   classNameBindings: ['labelType'],
12   for: readOnly('switchId'),
13   isVisible: readOnly('show'),
14   isAwesome: true,
15
16   labelValue: computed('isAwesome', function() {
17     return this.get('isAwesome') ?
18       'awesome-label' : 'lesser-awesome-label';
19   }),
20
21   type: computed('value', {
22     get() {
23       return this.get('value') ? 'on' : 'off';
24     }
25   }),
26
27   click(e) {
28     e.stopPropagation();
29     e.preventDefault();
30     this.sendToggle(this.get('value'));
31   }
32 });

```

```

1  {{#if hasBlock}}
2    {{yield label}}
3  {{else}}
4    {{label}}
5  {{/if}}
6

```



```

1  <ToggleLabel
2    @label={{this.labelVal}}
3    @value=false
4    sendToggle={{this.sendToggle}}
5  />

```

customOffLabel

// Octane Component syntax

```

1  import Component from '@glimmer/component';
2  import { tracked } from '@glimmer/tracking';
3  import { action } from '@ember/object';
4
5  export default class ToggleLabel extends Component {
6    @tracked isAwesome = true;
7
8    get type() {
9      return this.args.value ? 'on' : 'off';
10   }
11
12   get labelValue() {
13     return this.isAwesome ?
14       'awesome-label' : 'lesser-awesome-label';
15   }
16
17   @action
18   handleClick(e) {
19     e.stopPropagation();
20     e.preventDefault();
21     this.args.sendToggle(this.args.value);
22   }
23 }
24

```

```

1  <label
2    for="{{@switchId}}"
3    {{on 'click' this.handleClick}}
4
5    class="
6      toggle-text
7      toggle-prefix
8      {{this.type}}-label
9      {{if @show 'is-visible' 'is-hidden'}}
10   "
11  >
12   {{#if @hasBlock}}
13     {{yield @label}}
14   {{else}}
15     {{@label}}
16   {{/if}}
17 </label>
18

```





```

1 {{x-toggle-label
2   label=labelVal
3   value=false
4   sendToggle=(action 'sendToggle')
5 }}

```

customOffLabel

// Classic Component syntax

```

1 import { readOnly } from '@ember/object/computed';
2 import Component from '@ember/component';
3 import { computed } from '@ember/object';
4 import layout from './template';
5
6 export default Component.extend({
7   layout,
8   tagName: 'label',
9   attributeBindings: ['for'],
10  classNames: ['toggle-text', 'toggle-prefix'],
11  classNameBindings: ['labelType'],
12  for: readOnly('switchId'),
13  isVisible: readOnly('show'),
14  isAwesome: true,
15
16  labelValue: computed('isAwesome', function() {
17    return this.get('isAwesome') ?
18      'awesome-label' : 'lesser-awesome-label';
19  }),
20
21  type: computed('value', {
22    get() {
23      return this.get('value') ? 'on' : 'off';
24    }
25  }),
26
27  click(e) {
28    e.stopPropagation();
29    e.preventDefault();
30    this.sendToggle(this.get('value'));
31  }
32 });

```

```

1 {{#if hasBlock}}
2   {{yield label}}
3 {{else}}
4   {{label}}
5 {{/if}}
6

```



```

1 <ToggleLabel
2   @label={{this.labelVal}}
3   @value=false
4   sendToggle={{this.sendToggle}}
5 />

```

customOffLabel

// Octane Component syntax

```

1 import Component from '@glimmer/component';
2 import { tracked } from '@glimmer/tracking';
3 import { action } from '@ember/object';
4
5 export default class ToggleLabel extends Component {
6   @tracked isAwesome = true;
7
8   get type() {
9     return this.args.value ? 'on' : 'off';
10  }
11
12  get labelValue() {
13    return this.isAwesome ?
14      'awesome-label' : 'lesser-awesome-label';
15  }
16
17  @action
18  handleClick(e) {
19    e.stopPropagation();
20    e.preventDefault();
21    this.args.sendToggle(this.args.value);
22  }
23 }
24

```

```

1 <label
2   for="{{@switchId}}"
3   {{on 'click' this.handleClick}}
4
5   class="
6     toggle-text
7     toggle-prefix
8     {{this.type}}-label
9     {{if @show 'is-visible' 'is-hidden'}}
10  "
11 >
12   {{#if @hasBlock}}
13     {{yield @label}}
14   {{else}}
15     {{@label}}
16   {{/if}}
17 </label>
18

```





```

1  {{x-toggle-label
2    label=labelVal
3    value=false
4    sendToggle=(action 'sendToggle')
5  }}

```

customOffLabel

// Classic Component syntax

```

1  import { readOnly } from '@ember/object/computed';
2  import Component from '@ember/component';
3  import { computed } from '@ember/object';
4  import layout from './template';
5
6  export default Component.extend({
7    layout,
8    tagName: 'label',
9    attributeBindings: ['for'],
10   classNameBindings: ['toggle-text', 'toggle-prefix'],
11   classNameBindings: ['labelType'],
12   for: readOnly('switchId'),
13   isVisible: readOnly('show'),
14   isAwesome: true,
15
16   labelValue: computed('isAwesome', function() {
17     return this.get('isAwesome') ?
18       'awesome-label' : 'lesser-awesome-label';
19   }),
20
21   type: computed('value', {
22     get() {
23       return this.get('value') ? 'on' : 'off';
24     }
25   }),
26
27   click(e) {
28     e.stopPropagation();
29     e.preventDefault();
30     this.sendToggle(this.get('value'));
31   }
32 });

```

```

1  {{#if hasBlock}}
2    {{yield label}}
3  {{else}}
4    {{label}}
5  {{/if}}
6

```



```

1  <ToggleLabel
2    @label={{this.labelVal}}
3    @value=false
4    sendToggle={{this.sendToggle}}
5  />

```

customOffLabel

// Octane Component syntax

```

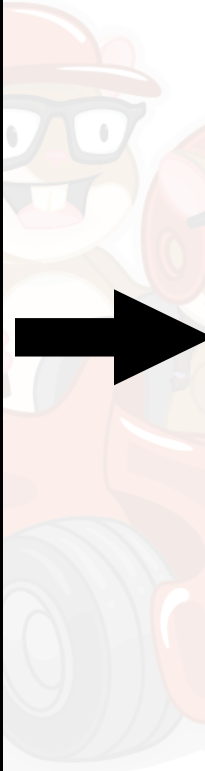
1  import Component from '@glimmer/component';
2  import { tracked } from '@glimmer/tracking';
3  import { action } from '@ember/object';
4
5  export default class ToggleLabel extends Component {
6    @tracked isAwesome = true;
7
8    get type() {
9      return this.args.value ? 'on' : 'off';
10   }
11
12   get labelValue() {
13     return this.isAwesome ?
14       'awesome-label' : 'lesser-awesome-label';
15   }
16
17   @action
18   handleClick(e) {
19     e.stopPropagation();
20     e.preventDefault();
21     this.args.sendToggle(this.args.value);
22   }
23 }
24

```

```

1  <label
2    for="{{@switchId}}"
3    {{on 'click' this.handleClick}}
4
5    class="
6      toggle-text
7      toggle-prefix
8      {{this.type}}-label
9      {{if @show 'is-visible' 'is-hidden'}}
10   "
11  >
12   {{#if @hasBlock}}
13     {{yield @label}}
14   {{else}}
15     {{@label}}
16   {{/if}}
17 </label>
18

```





```

1  {{x-toggle-label
2    label=labelVal
3    value=false
4    sendToggle=(action 'sendToggle')
5  }}

```

customOffLabel

// Classic Component syntax

```

1  import { readOnly } from '@ember/object/computed';
2  import Component from '@ember/component';
3  import { computed } from '@ember/object';
4  import layout from './template';
5
6  export default Component.extend({
7    layout,
8    tagName: 'label',
9    attributeBindings: ['for'],
10   classNames: ['toggle-text', 'toggle-prefix'],
11   classNameBindings: ['labelType'],
12   for: readOnly('switchId'),
13   isVisible: readOnly('show'),
14   isAwesome: true,
15
16   labelValue: computed('isAwesome', function() {
17     return this.get('isAwesome') ?
18       'awesome-label' : 'lesser-awesome-label';
19   }),
20
21   type: computed('value', {
22     get() {
23       return this.get('value') ? 'on' : 'off';
24     }
25   }),
26
27   click(e) {
28     e.stopPropagation();
29     e.preventDefault();
30     this.sendToggle(this.get('value'));
31   }
32 });

```

```

1  {{#if hasBlock}}
2    {{yield label}}
3  {{else}}
4    {{label}}
5  {{/if}}
6

```



```

1  <ToggleLabel
2    @label={{this.labelVal}}
3    @value=false
4    sendToggle={{this.sendToggle}}
5  />

```

customOffLabel

// Octane Component syntax

```

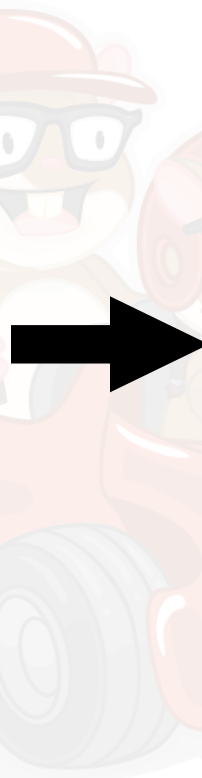
1  import Component from '@glimmer/component';
2  import { tracked } from '@glimmer/tracking';
3  import { action } from '@ember/object';
4
5  export default class ToggleLabel extends Component {
6    @tracked isAwesome = true;
7
8    get type() {
9      return this.args.value ? 'on' : 'off';
10   }
11
12   get labelValue() {
13     return this.isAwesome ?
14       'awesome-label' : 'lesser-awesome-label';
15   }
16
17   @action
18   handleClick(e) {
19     e.stopPropagation();
20     e.preventDefault();
21     this.args.sendToggle(this.args.value);
22   }
23 }
24

```

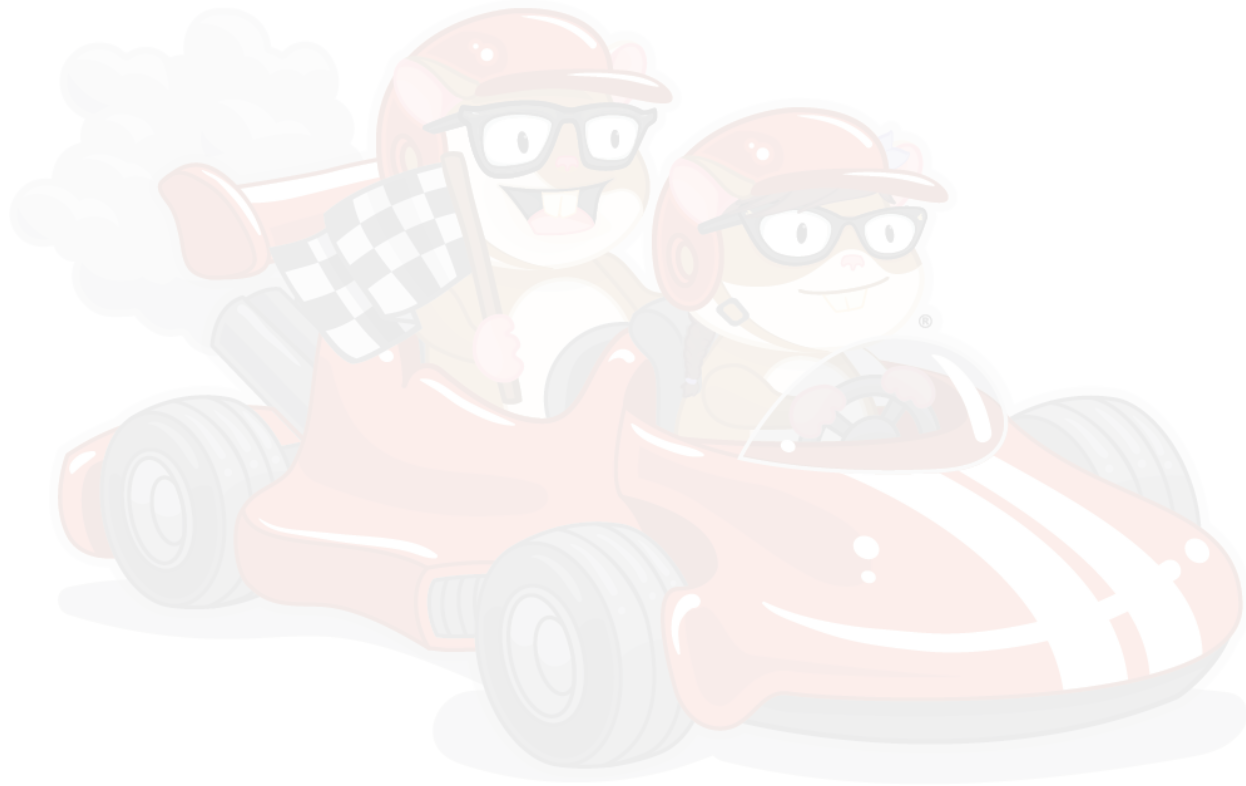
```

1  <label
2    for="{{@switchId}}"
3    {{on 'click' this.handleClick}}
4
5    class="
6      toggle-text
7      toggle-prefix
8      {{this.type}}-label
9      {{if @show 'is-visible' 'is-hidden'}}
10   "
11  >
12   {{#if @hasBlock}}
13     {{yield @label}}
14   {{else}}
15     {{@label}}
16   {{/if}}
17 </label>
18

```



Tools & Goodness



Tools & Goodness

- Ember Codemods: <https://github.com/ember-codemods>
- Ember Atlas: <https://www.notion.so/The-Ember-Atlas-4094f81c86c34badb4a562ed29414ae1>
- The Octane Edition of Ember:
<https://emberjs.com/editions/octane/>
- Ember Blog - Octane is Here:
<https://blog.emberjs.com/2019/12/20/octane-is-here.html>

Tools & Goodness

- Ember Codemods: <https://github.com/ember-codemods>
- Ember Atlas: <https://www.notion.so/The-Ember-Atlas-4094f81c86c34badb4a562ed29414ae1>
- The Octane Edition of Ember:
<https://emberjs.com/editions/octane/>
- Ember Blog - Octane is Here:
<https://blog.emberjs.com/2019/12/20/octane-is-here.html>

Thank you!

