



Accessibility-flavored

# React components

make your design  
system delicious

Kathleen McMahon  
[@resource11](https://twitter.com/resource11)

**<https://noti.st/resource11>**

Slide deck posted after the talk

# @resource11

[Twitter](#) | [Instagram](#) | [GitHub](#)



**Good grief,  
there's an  
agenda...**

Why accessibility first?

Design systems are a cookbook

Design systems and React

Icons

Buttons

Inputs

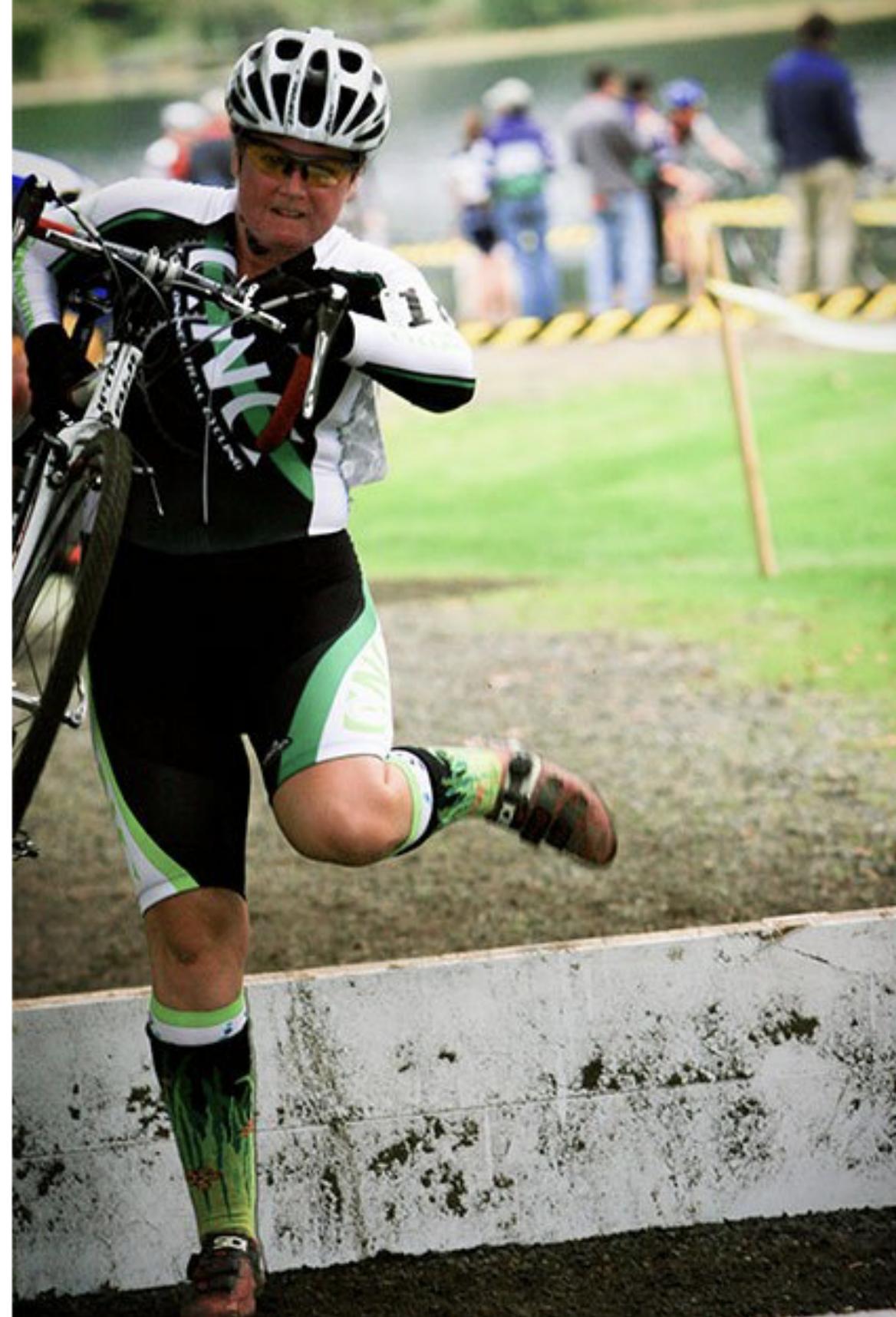
Documentation

# Who are you, again?



carGurus®

# Cyclocross!









@resource11

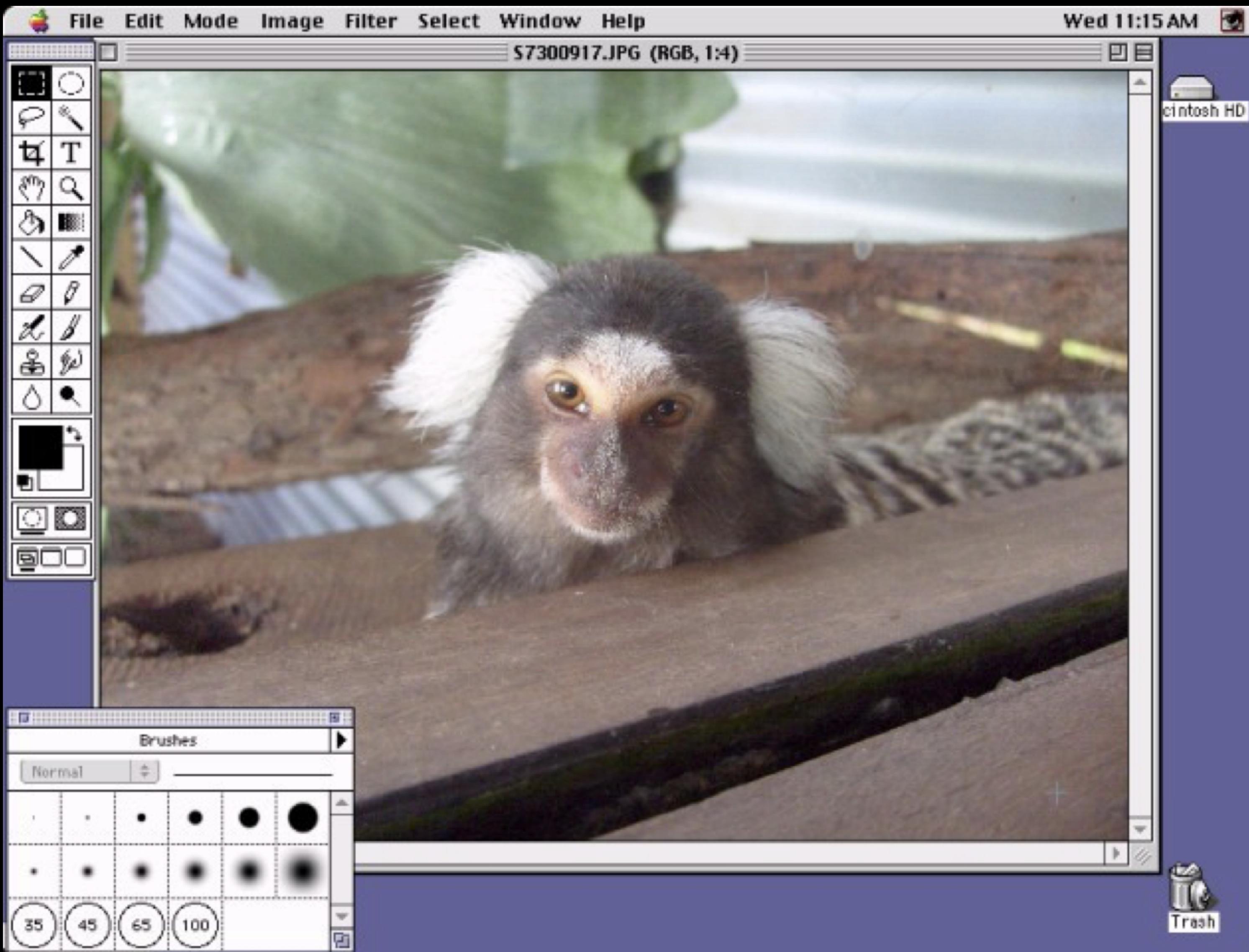
# Dev Dinosaur

Disney Pixar  
**THE GOOD  
DINOSAUR**



@resource11

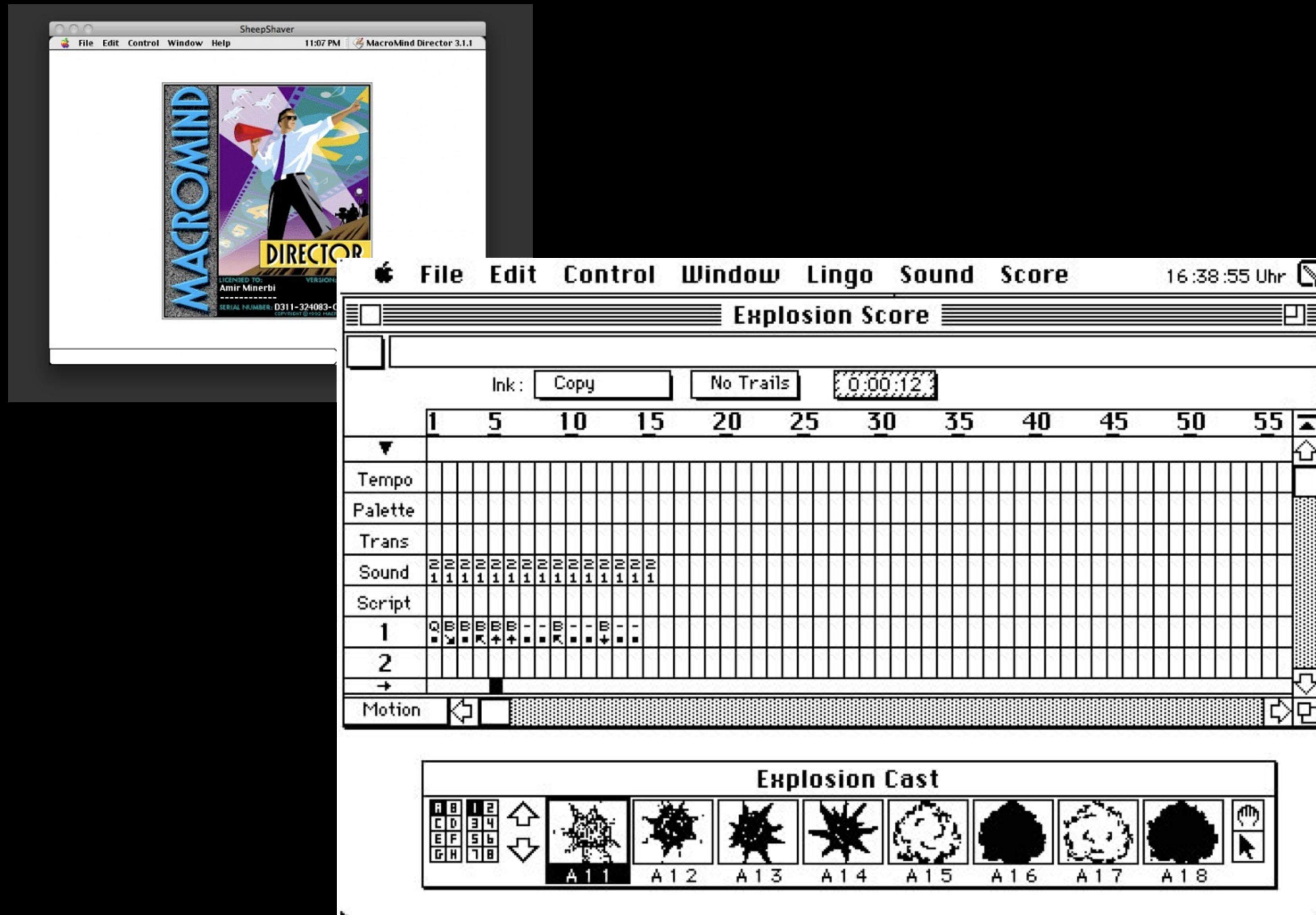




@resource11



@resource11





@resource11

Welcome to  
Zip™ drive  
Inside is all the stuff you  
to get started





@resource11





NETSCAPE



# HTML + CSS + JS

# JavaScript Framework of the week

A photograph of a group of people at a carnival or amusement park. In the foreground, a man wearing a red baseball cap and a blue patterned shirt is looking towards the camera with a wide-open mouth, appearing to shout or sing. Behind him, several other people are visible, some smiling and looking towards the camera. In the background, a large roller coaster with red and yellow tracks and a red and white marquee sign is visible against a dark sky.

**OPEN WIDE!!**

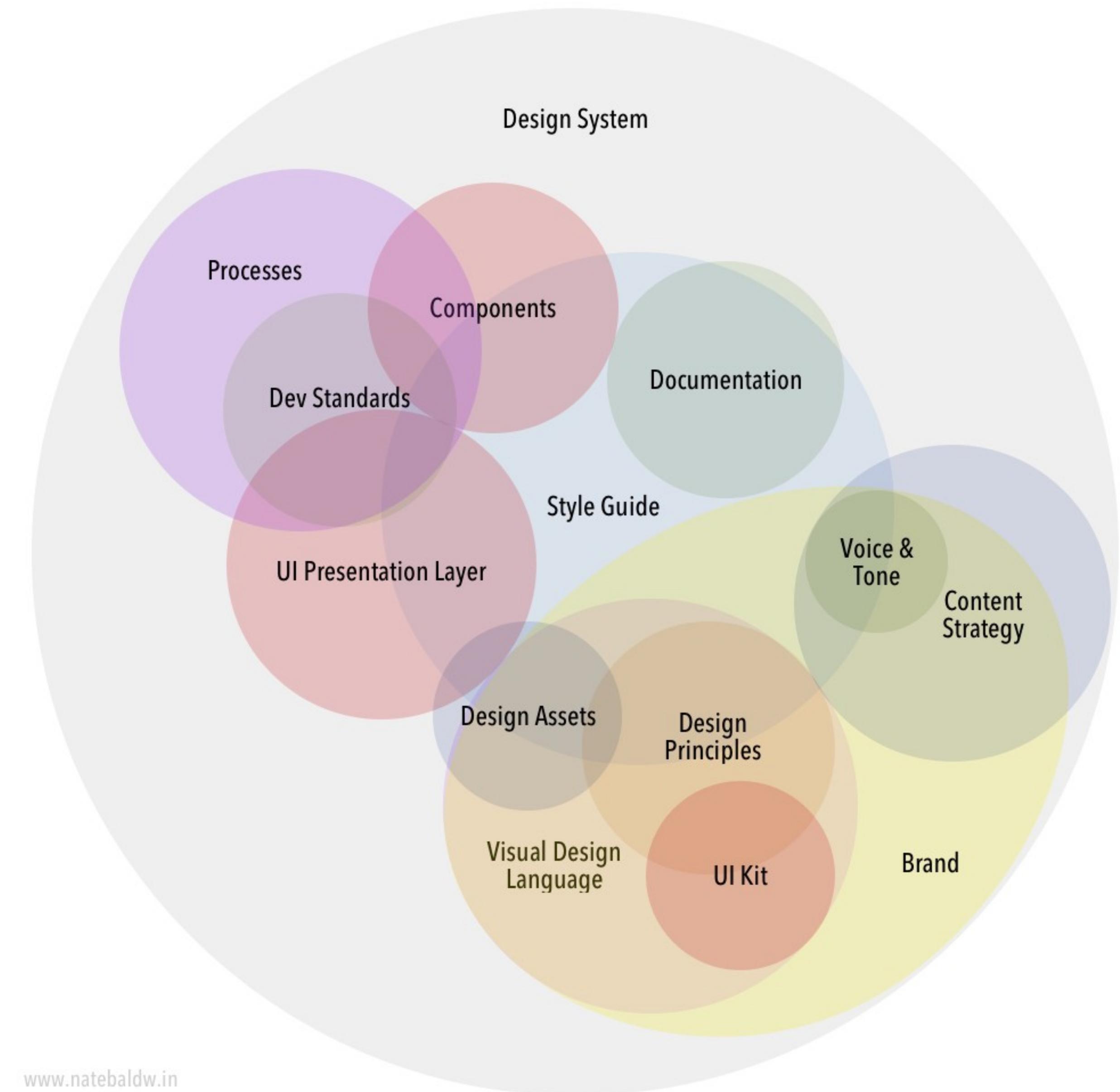
A man with short brown hair, wearing a black tuxedo jacket over a white shirt, is sitting on a blue folding chair. He is looking off to his right with a neutral expression. The background is a dense green jungle with sunlight filtering through the leaves.

Dinosaurs are  
**ALWAYS** the  
hotness

O'REILLY®



@resource11



Great design systems  
combine team AND  
consumer experience



First Priority

**Business logic out,  
accessibility in**

# Why accessibility first?

---

# Our users are diverse

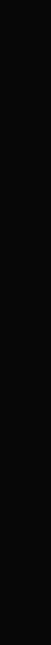
Vision



Hearing



Motor



Sensory

Language

Low bandwidth



February 2019

# WebAIM Million

An accessibility analysis of the top  
1,000,000 home pages

97.8%

of home pages had detectable  
WCAG 2 failures.

59%

of 3.4 million form inputs  
were unlabeled



60.1%

of the 1 million pages had  
ARIA present



# **Web Content Accessibility Guidelines (WCAG)**



Imagine... accessibility  
baked into components.



@resource11

Your Design  
system is a  
cookbook

Cook Book



# Cookbooks have personality

**The Personality of a Cookbook**

posed of known and stable ingredients with unknown and elusive ones to make a  
ture as familiar, friendly and exhilarating as a pine woods early on a summer morn

**The Stable Ingredients**

homemaker needs for herself and for her family, based on a sound knowledge of what is  
on preparing and serving foods, based on an easy handling of all fundamental facts

**The Elusive Charm**

publication, and on the presentation of those facts in a simple, concise,  
organized, specific, easy-to-follow, step-by-step procedure that gives

**This Beauty**

feeling of confidence in the final product.  
—a sense of humor—a flair for the clever idea  
something called style, but above all a feeling for the kind of beauty that comes from clear overtones: a light touch

**This Beauty**

has been made possible by the cooperation and confidence  
food and equipment industry, whose constant flow of inspiring  
about them in their work-a-day world.



@resource11

## BAKING

Yeast breads	Temp.	Time in minutes
Loaves (8½ x 4½ x 2½)	375°-400°	30-45
Rolls Plain Sweet	400° 375°-400°	10-20 10-25

## Quick breads

Biscuits	450°	10-15
Muffins	375°-425°	15-25
Corn bread	425°	12-15
Popovers then	475° 350°	15 25-30
Nut bread	350°	40-60
Coffeecake	350°-375°	25-40

## Pastry

Pastry shell	450°	10-12
Fruit pies or then	400° 450° 350°	35-50 10 30
Custard pie	400°	25-30
Pumpkin pie	400°	50
Meringue	350°	12-15

## Cakes

Cakes with shortening		
Cupcakes	375°	20
Layers	350°-375°	20-35
Squares, oblongs, upside-down cakes	350°	25-40
Loaf cakes	300°-350°	45-85
Angel (tube)	375°	30-40
Chiffon (tube)	325°	55
then	350°	10-15
Sponge (tube)	325°	55-60
Fruitcake	275°	2½-3½ hrs.

## Cookies

Drop	350°-400°	8-15
Rolled	350°-375°	5-12
Refrigerator	350°-400°	8-15
Filled	350°-375°	8-12
Bars	325°-375°	10-35

## DEEP-FAT FRYING

	Temp.	Time in of fat
Croquettes	315°	till golden
Fritters French fr. s	375°	3-4
then	360°	till golden
Califlowerets	375°	3-5
Eggplant	375°	3-4
Onion rings	375°	2-5
Sea foods	375°	till golden
	375°	till golden
	375°	2

# Why nearly 9,000,000 women cherish this cook book



Top: Perfect Divinity, Toffee Bars, Caesar Salad, Spice Layer Cake; Bottom: Oval

You serve nutritious meals. This cook book lists calorie ratings of many foods; suggests plans for delicious meals.

You save time, effort, fuss. Here are techniques for fast efficient cooking; see the complete chapter on jiffy meals.

You reduce food bills. Recipes call for even cans and packages of ingredients—no waste.

You're a good hostess! See the section on entertaining—luncheons, brunches, buffets, table settings, centerpieces.

Every recipe in the new Cook Book is Test-Kitchen endorsed!

The checked seal means that each recipe has been tested over and over till it rates high in practicality, family appeal, and downright deliciousness! This red and white symbol assures you a perfect product every time! A recipe must meet our high standards and merit the faith of our readers.

These recipes are so practical—fast to fix, use on-hand ingredients, no special equipment. Even company dishes take minimum bother.

**Better Homes  
and Gardens  
TEST  
KITCHEN**

Better Homes & Gardens New Cook Book





@resource11



# Meats, poultry, and fish

## Meat

Methods of cooking 236-237

### Beef

Beefsteak and Kidney Pie 265  
 Beef-and-beef Dinner 245  
 Beef Patties, Saucy 264  
 238-239

Steak  
 Broiled  
 Flank  
 Minute  
 Pan-broiled  
 Round  
 Swiss  
 Stew, Old-fashioned  
 Stroganoff

### Pork

250  
 249  
 246-247

242  
 242  
 242  
 243  
 243  
 244  
 245

Rotisserie-roasted Ham 251  
 Sliced Buffet Ham 252  
 Picnic Roasts 251, 252  
 Crown Loin 248  
 Stuffed Shoulder 248  
 Smoked Shoulder Butt 248  
 Spareribs 248  
 Steaks with Apple 251  
 Sweet-sour Pork 249  
 Tenderloin 250

### Veal

Baked Veal-and-Ham Birds 256  
 Roast, Roast Stuffed 256  
 256  
 256  
 257

	OVEN	ROASTING	STEAMING	BOILING	TOTAL
Meat, poultry	150°-375°	150°-375°	150°-375°	150°-375°	150°-375°
Steak, broiled	325°	325°	325°	325°	325°
Flank, minute	325°	325°	325°	325°	325°
Pork, round	325°	325°	325°	325°	325°
Round, swiss	325°	325°	325°	325°	325°
Stew, old-fashioned	325°	325°	325°	325°	325°
Stroganoff	325°	325°	325°	325°	325°
Pork, 1 lb.	325°	325°	325°	325°	325°
Pork, 2 lbs.	325°	325°	325°	325°	325°
Pork, 3 lbs.	325°	325°	325°	325°	325°
Pork, 4 lbs.	325°	325°	325°	325°	325°
Pork, 5 lbs.	325°	325°	325°	325°	325°
Pork, 6 lbs.	325°	325°	325°	325°	325°
Pork, 7 lbs.	325°	325°	325°	325°	325°
Pork, 8 lbs.	325°	325°	325°	325°	325°
Pork, 9 lbs.	325°	325°	325°	325°	325°
Pork, 10 lbs.	325°	325°	325°	325°	325°
Pork, 11 lbs.	325°	325°	325°	325°	325°
Pork, 12 lbs.	325°	325°	325°	325°	325°
Pork, 13 lbs.	325°	325°	325°	325°	325°
Pork, 14 lbs.	325°	325°	325°	325°	325°
Pork, 15 lbs.	325°	325°	325°	325°	325°
Pork, 16 lbs.	325°	325°	325°	325°	325°
Pork, 17 lbs.	325°	325°	325°	325°	325°
Pork, 18 lbs.	325°	325°	325°	325°	325°
Pork, 19 lbs.	325°	325°	325°	325°	325°
Pork, 20 lbs.	325°	325°	325°	325°	325°
Pork, 21 lbs.	325°	325°	325°	325°	325°
Pork, 22 lbs.	325°	325°	325°	325°	325°
Pork, 23 lbs.	325°	325°	325°	325°	325°
Pork, 24 lbs.	325°	325°	325°	325°	325°
Pork, 25 lbs.	325°	325°	325°	325°	325°
Pork, 26 lbs.	325°	325°	325°	325°	325°
Pork, 27 lbs.	325°	325°	325°	325°	325°
Pork, 28 lbs.	325°	325°	325°	325°	325°
Pork, 29 lbs.	325°	325°	325°	325°	325°
Pork, 30 lbs.	325°	325°	325°	325°	325°
Pork, 31 lbs.	325°	325°	325°	325°	325°
Pork, 32 lbs.	325°	325°	325°	325°	325°
Pork, 33 lbs.	325°	325°	325°	325°	325°
Pork, 34 lbs.	325°	325°	325°	325°	325°
Pork, 35 lbs.	325°	325°	325°	325°	325°
Pork, 36 lbs.	325°	325°	325°	325°	325°
Pork, 37 lbs.	325°	325°	325°	325°	325°
Pork, 38 lbs.	325°	325°	325°	325°	325°
Pork, 39 lbs.	325°	325°	325°	325°	325°
Pork, 40 lbs.	325°	325°	325°	325°	325°
Pork, 41 lbs.	325°	325°	325°	325°	325°
Pork, 42 lbs.	325°	325°	325°	325°	325°
Pork, 43 lbs.	325°	325°	325°	325°	325°
Pork, 44 lbs.	325°	325°	325°	325°	325°
Pork, 45 lbs.	325°	325°	325°	325°	325°
Pork, 46 lbs.	325°	325°	325°	325°	325°
Pork, 47 lbs.	325°	325°	325°	325°	325°
Pork, 48 lbs.	325°	325°	325°	325°	325°
Pork, 49 lbs.	325°	325°	325°	325°	325°
Pork, 50 lbs.	325°	325°	325°	325°	325°
Pork, 51 lbs.	325°	325°	325°	325°	325°
Pork, 52 lbs.	325°	325°	325°	325°	325°
Pork, 53 lbs.	325°	325°	325°	325°	325°
Pork, 54 lbs.	325°	325°	325°	325°	325°
Pork, 55 lbs.	325°	325°	325°	325°	325°
Pork, 56 lbs.	325°	325°	325°	325°	325°
Pork, 57 lbs.	325°	325°	325°	325°	325°
Pork, 58 lbs.	325°	325°	325°	325°	325°
Pork, 59 lbs.	325°	325°	325°	325°	325°
Pork, 60 lbs.	325°	325°	325°	325°	325°
Pork, 61 lbs.	325°	325°	325°	325°	325°
Pork, 62 lbs.	325°	325°	325°	325°	325°
Pork, 63 lbs.	325°	325°	325°	325°	325°
Pork, 64 lbs.	325°	325°	325°	325°	325°
Pork, 65 lbs.	325°	325°	325°	325°	325°
Pork, 66 lbs.	325°	325°	325°	325°	325°
Pork, 67 lbs.	325°	325°	325°	325°	325°
Pork, 68 lbs.	325°	325°	325°	325°	325°
Pork, 69 lbs.	325°	325°	325°	325°	325°
Pork, 70 lbs.	325°	325°	325°	325°	325°
Pork, 71 lbs.	325°	325°	325°	325°	325°
Pork, 72 lbs.	325°	325°	325°	325°	325°
Pork, 73 lbs.	325°	325°	325°	325°	325°
Pork, 74 lbs.	325°	325°	325°	325°	325°
Pork, 75 lbs.	325°	325°	325°	325°	325°
Pork, 76 lbs.	325°	325°	325°	325°	325°
Pork, 77 lbs.	325°	325°	325°	325°	325°
Pork, 78 lbs.	325°	325°	325°	325°	325°
Pork, 79 lbs.	325°	325°	325°	325°	325°
Pork, 80 lbs.	325°	325°	325°	325°	325°
Pork, 81 lbs.	325°	325°	325°	325°	325°
Pork, 82 lbs.	325°	325°	325°	325°	325°
Pork, 83 lbs.	325°	325°	325°	325°	325°
Pork, 84 lbs.	325°	325°	325°	325°	325°
Pork, 85 lbs.	325°	325°	325°	325°	325°
Pork, 86 lbs.	325°	325°	325°	325°	325°
Pork, 87 lbs.	325°	325°	325°	325°	325°
Pork, 88 lbs.	325°	325°	325°	325°	325°
Pork, 89 lbs.	325°	325°	325°	325°	325°
Pork, 90 lbs.	325°	325°	325°	325°	325°
Pork, 91 lbs.	325°	325°	325°	325°	325°
Pork, 92 lbs.	325°	325°	325°	325°	325°
Pork, 93 lbs.	325°	325°	325°	325°	325°
Pork, 94 lbs.	325°	325°	325°	325°	325°
Pork, 95 lbs.	325°	325°	325°	325°	325°
Pork, 96 lbs.	325°	325°	325°	325°	325°
Pork, 97 lbs.	325°	325°	325°	325°	325°
Pork, 98 lbs.	325°	325°	325°	325°	325°
Pork, 99 lbs.	325°	325°	325°	325°	325°
Pork, 100 lbs.	325°	325°	325°	325°	325°
Pork, 101 lbs.	325°	325°	325°	325°	325°
Pork, 102 lbs.	325°	325°	325°	325°	325°
Pork, 103 lbs.	325°	325°			



@resource11

# Soups and sauces

## Soups<sup>s</sup>

<b>Soups</b>	
Accompaniments for soups	333
Bean Soup	331
Borsch, Russian	330
Borsch, Sparkling	334
Broth (Bouillon)	330
Canned-soup combinations	333
Cream of Chicken-Mushroom	333
Creole Clam Bisque	333
Pea Soup Royale	333
Mongole	333
Cream of Potato Soup	331
Cream of Tomato Soup	331
Cucumber Soup	334
Garnishes for soups	333
Gazpacho, Spanish	334
Onion Soup, French	330
Oyster Stew	332
Pacific Chowder	333
Split-pea Soup	331
Stock	330
Brown	330
White	330
Tomato soups	330
Cream of Tomato Soup	331
Puree Mongole	333
Tomato Bouillon	333
Tomato Chili Stew	334
Tomato-Rice Soup	334
Unshake	331
Cherry Sauce	338
Cranberry Sauce	335
Cranberry Sauce, Tad	335
Creole Sauce	335
Cucumber Sauce	335
Cumberland Sauce	335
Dill Sauce	336
Garlic Butter	335
Gourmet Sauce	335
Herb Butter	335
Hollandaise	338
Classic	338
Easy	338
Jiffy	338
Horseradish Sauces	336
Lemon Butter	335
Mint Sauces for Lamb	336
Mushroom Sauce	335
Mushroom Topper, Quick	335
Mushroom Topper, Creamy	336
Mustard Sauces	337
Pickle Sauce	336
Sauce for Ham	337

# Soups and sauces

# Chapter 17

soups and sal

Ver.  
Exhibit  
Stan  
B.

**ROASTING**  
Tom  
ding roasts  
eef  
Rare  
Medium  
Well-done

325° 325° 325° 325° 325°

32 40  
35 40  
35 40

ressour

ce11

# **THERMOMETER READINGS FOR MEATS**

Temp. of meat

Beef  
Rare  
Medium  
Well-done  
Veal  
Lamb  
Fresh pork  
Turkey  
In open  
In foil

heat  
age

195°  
190°

1 time  
minutes

**BROILING**

Beef steaks\*  
1-inch  
sare.

French  
Rare  
Sodium

1

all time  
minutes

10  
12

12-2

# Vegetables

# Chapter 18

pages 339 through 368

<b>Vegetables</b>	
Cooking chart	341-343
Cooking methods	340
Creamed Vegetables	340
Scalloped Vegetables	340
Vegetables au Gratin	340
Vegetables Cooked in Oil	340
Artichokes	341, 344
Artichokes with Butter	344
Crab-stuffed Artichokes	344
Asparagus	341, 345
Asparagus with Croutons	344
Asparagus dress-ups	344
Beans	341
Green Beans	346
Pinto Beans	346
Cabbage	341
Cabbage with Cheese	346
Cabbage with Golden Sauce	347
Savory Cabbage	347
Pennsylvania Red Cabbage	347
Carrots	342
Carrot Bake with Basil	348
Carrots Chinese	348
Glazed Carrots	348
Herbed Carrots	348
Minted Carrots	348
Cauliflower	342
Cheese-frosted Cauliflower	348
with Shrimp Sauce	348
Celeriac	342
Celery	342
Celery Luncheon Bake	349
Celery Victor	349
N-M Celery Oriental	349
Corn	342
on the Cob	349
Greens	342
Kohlrabi	342
Leeks	342
Mushrooms	342
Creamed Mushrooms	350
Fresh Mushroom Saute	350
Okra	343
Onions	350
Creamed Onions	351
Dutch Glazed Onions	351
French-fried Onion Rings	350
Onions Gourmet	343, 351
Parsnips	343, 351
Peas	351
Company Peas	343
Garden Peas in Cream	343
Peas Cooked in Oil	343
Potatoes	351
Baked Potatoes	343
Boiled Potatoes	343
Mashed Potatoes	343
Potato Fries	343
see cooking chart	342



@resource11

**18 Vegetables**

**ROASTING**

	Temp.	Min. per pound
Standing roasts		
Beef	325°	20-22
Rare	325°	24-27
Medium	325°	29-32
Well-done	325°	35-40
Veal	325°	30-35
Lamb	325°	35-40
Fresh pork	325°	35-40
Loin or blade	325°	50-55*
Crown		
Ham — (See timetable on page 253 for specific times.)		
Rolled roasts (all)		
10 minutes more per pound		
	Total time	
	Temp.	

\*(See timetable on page 253 for specific times.)

**THERMOMETER READINGS FOR MEATS**

	Temp. of meat
Beef	140°
Rare	160°
Medium	170°
Well-done	170°
Veal	170-180°
Lamb	185°
Fresh pork	195°
Turkey	190°
In open pan	
In foil	

**BROILING**

	Total time in min.
Beef steaks*	
1-inch	
Rare	
Medium	
1½ -inch	
Rare	
Medium	
*(Place thin steaks not	
thick steaks not	
Lamb chops, steaks	
1-inch	
Medium	
Well-done	

**Table of Time**

Along dotted line

Cut along dotted line



@resource11

# Desserts

# chapter 10

pages 187 through 208

<b>Desserts</b>		189
<b>Alaskas</b>		
Baked Alaska	199	
Melba Alaska	199	
Ribbon Alaska Pie	199	
<b>Cake desserts</b>		
Almond-Brittle Cake	203	
Chocolate-fleck Cake	203	
Chocolate Fluff Layers	202	
Coffee Ribbon Loaf	203	
Daffodil Layers	202	
Fruited Angel Surprise	203	
Lemonade Angel Dessert	203	
Angel Ring	202	
Dessert	202	
<b>Eclairs</b>		
<b>Fruit desserts</b>		
Ambrosia	205	
Apple Betty Pie	204	
Apple Cobbler	207	
Apple Dumplings	204	
Apple-Raisin Rolls	206	
Apples, Baked	204	
Applesauce	204	
Blueberry Crumb Pudding	191	
Blueberry Rolls	206	
Cherry Puff	205	
Cherry Cobbler	207	
Water Fruit	204	
Fruit	205	
<b>Glorified Rice</b>		
<b>Ice cream</b>		
Stirred Custard	200	
Strawberry Floating Island	191	
Chocolate-Almond Velvet	197	
Chocolate-chip Cups	198	
Chocolate Ice Cream	196	
Coffee Ice Cream	196	
French Chocolate	196	
Ice Cream	198	
Holiday Ice Cream	196	
How to freeze ice cream	197	
Orange Ice Cream	197	
Peach-tree Ice Cream	197	
Peppermint Ice Cream	197	
Pineapple Ice Cream	197	
Pistachio Ice Cream	197	
Strawberry Ice Cream	198	
Tutti-frutti Tortoni	196	
Vanilla Custard Ice Cream	196	
Vanilla Ice Cream	200	
<b>Soufflés</b>		



# Table settings, and entertaining

Table settings  
Centerpieces..... 363, 368, 369

Tables..... 361-362

Special meals

Buffets and casual dinners

Before-the-Game Buffet..... 375

Card-table Buffet..... 375

Chopstick Dinner..... 375

Fireside Supper for 4..... 375

Happy Birthday Dinner..... 375

Potluck with a Plan..... 375

Spaghetti Supper..... 375

holidays..... 378

Snacks and refreshments

"Come on over for dessert"..... 376

Men's Choice Snack Bar..... 376

Planned Refrigerator Raid..... 376

Platter Party..... 376

Special-occasion Coffee..... 376

Summer Refreshment Plate..... 376

In Between..... 376

chapter 19  
Pages 369 through 378

Very slow oven..... 450°  
Slow oven..... 450°-525°  
Moderate oven..... 500°-525°  
Hot oven..... 500°  
Very hot oven..... 500°-525°  
Extremely hot oven..... 500°-525°

ROASTING

Standing roasts  
Beef..... 325°  
Rare..... 325°  
Medium..... 325°  
Well-done..... 325°

Veal..... 325°  
Lamb..... 325°

Fresh pork..... 325°  
Loin or blade..... 325°

Crown..... 325°  
Ham—(See timetable on page 253 for specific times)  
Rolled roasts (all)..... 10 minutes more per pound

Total time.....

Temp.....

Poultry  
Chicken..... 400°  
Duckling..... 325°  
Turkey (8-12 pound)\*..... 325°

Open pan..... 325°  
In foil..... 450°

\*(See timetable on page 272 for specific times)

THERMOMETER  
READINGS FOR MEATS

Temp. of meat

Beef  
Rare..... 140°  
Medium..... 160°

Well-done..... 170°

Veal..... 170°

Lamb..... 180°

Fresh pork..... 185°

Turkey..... 190°

In open pan..... 140°

In foil..... 160°

BROILING

Beef steaks\*  
1-inch..... 140°

Rare..... 160°

Medium..... 170°

Well-done..... 170°

Veal..... 170°

Lamb..... 180°

Fresh pork..... 185°

Turkey..... 190°

In open pan..... 140°

In foil..... 160°

19 Table settings  
and entertaining

20 Inc.

# Design Systems and React

# React

A JavaScript library for building user interfaces

[Get Started](#)[Take the Tutorial >](#)

## Declarative

React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes.

## Component-Based

Build encapsulated components that manage their own state, then compose them to make complex UIs.

Since component logic is written in JavaScript instead of templates, you can easily connect

## Learn Once, Write Anywhere

We don't make assumptions about the rest of your technology stack, so you can develop new features in React without rewriting existing code.

React can also render on the server using Node and on mobile using a React Native

# Accessibility

## Why Accessibility?

Web accessibility (also referred to as [a11y](#)) is the design and creation of websites that can be used by everyone. Accessibility support is necessary to allow assistive technology to interpret web pages.

React fully supports building accessible websites, often by using standard HTML techniques.

## Standards and Guidelines

### WCAG

The [Web Content Accessibility Guidelines](#) provides guidelines for creating accessible web sites.

INSTALLATION ▾

MAIN CONCEPTS ▾

ADVANCED GUIDES ^

Accessibility

Code-Splitting

Context

Error Boundaries

Forwarding Refs

Fragments

Higher-Order Components

Integrating with Other Libraries

JSX In Depth

Optimizing Performance

Portals

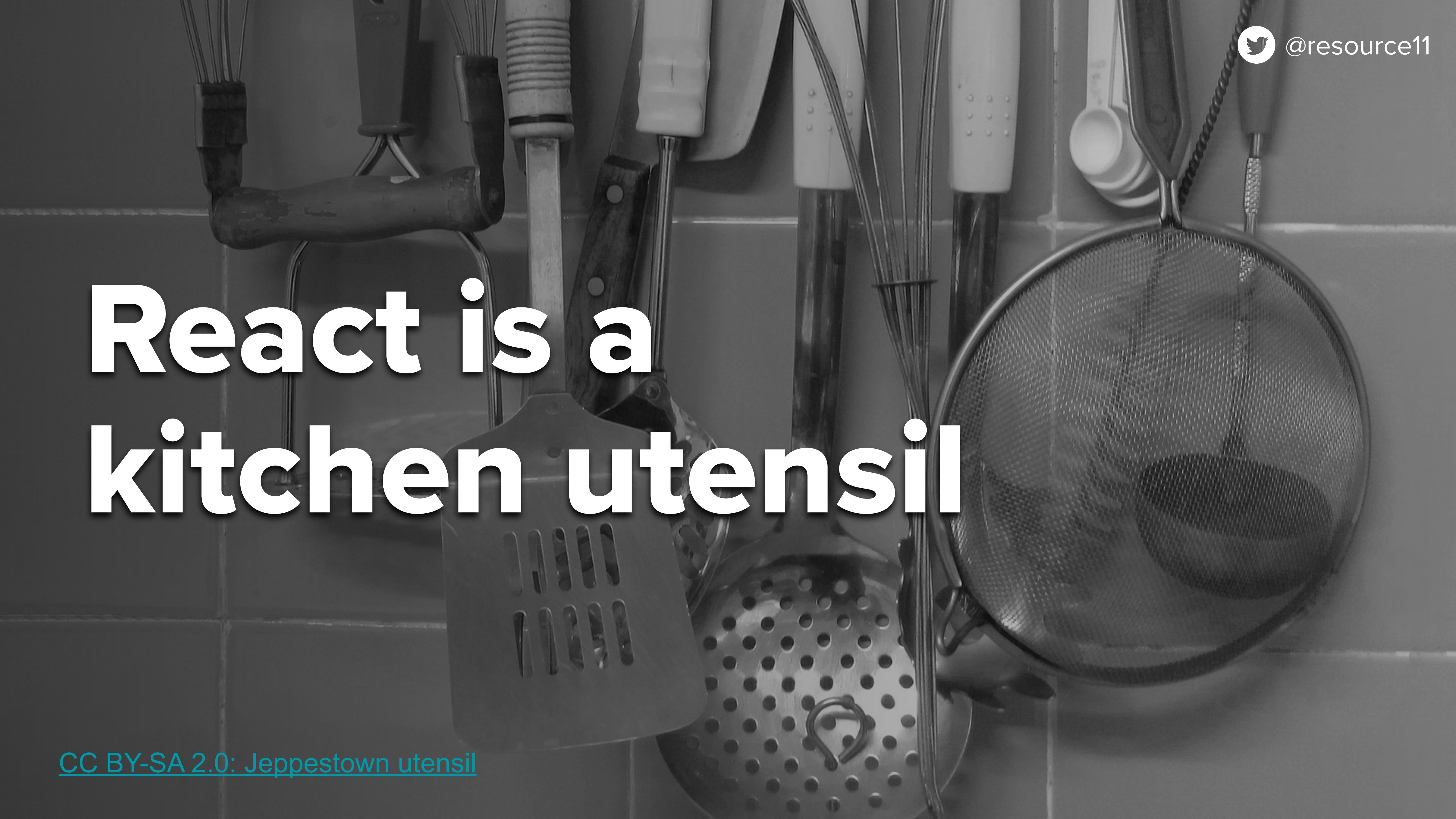
React Without ES6

React Without JSX

Reconciliation

Refs and the DOM

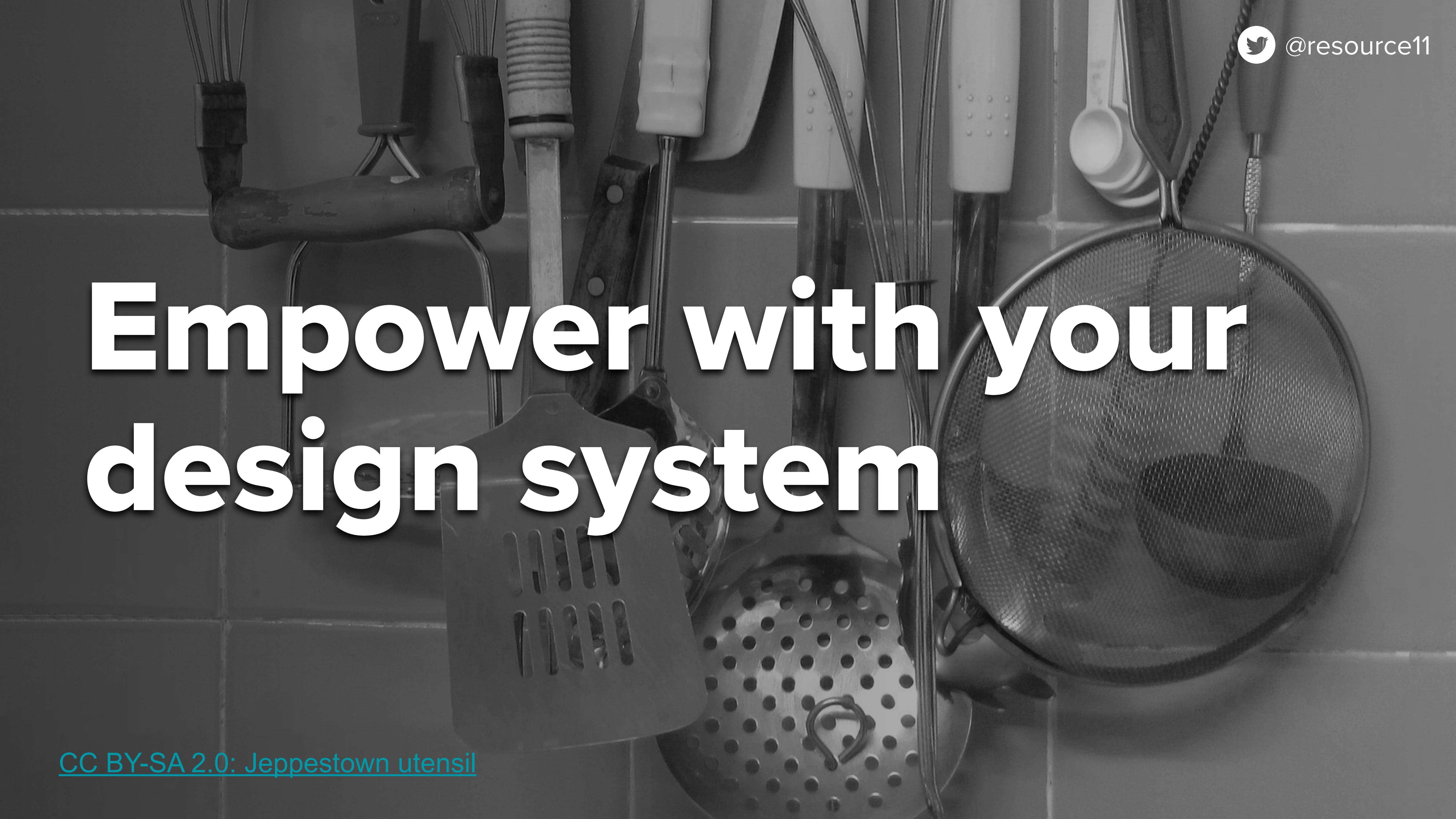
Render Props



React is a  
kitchen utensil



You are the cook



Empower with your  
design system

**Matt May**

@mattmay

▼

Getting a hard-fought accessibility win is nice. But even nicer is seeing a co-worker doing good accessibility work without you having to ask (much less fight) for it.

6:43 PM · Nov 6, 2019 · [TweetDeck](#)

---

**32** Retweets**162** Likes



@resource11

Components are  
your tried and  
true recipes

RUTH WAKEFIELD'S  
TOLL HOUSE

Tried and True Recipes

RUTH GRAVES  
WAKEFIELD  
Dietitian • Lecturer



M. BARROWS & COMPANY, INC.  
New York  
1941



@resource11

# WCAG is your reference material

**new  
revised  
edition**

The image shows the front cover of a book. The title "CAG is Your Encyclopedia CookBook" is written in large, white, serif capital letters across the center. Above the title, the text "CULINARY ARTS INSTITUTE" is printed in smaller, black, serif capital letters. In the top left corner, there is a decorative illustration of various kitchen utensils and ingredients, including a whisk, a rolling pin, and a bowl of fruit. The background of the cover is a light beige color.

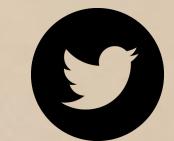
# materia

# Component recipe

*Luncheon or dinner guests are sure to enjoy colorful Citrus Lobster Salad.*

CITRUS LOBSTER SALAD

# FRESH SPRINGTIME SALAD



@resource11

1 pound fresh asparagus spears  
1 cup torn fresh spinach  
2 cups torn red leaf lettuce  
2 cups torn leaf lettuce  
1/4 cup Warm Seasoned Dressing

Snap off tough ends of asparagus; trim scales with a knife. Cut asparagus into 1-inch pieces. Place in a large bowl of cold water. Cover and refrigerate until crisp-tender.

Combine asparagus and warm seasoned dressing in a large bowl, tossing gently. Pour dressing over salad; toss gently. Serve immediately. Yield: 9 servings (30 calories per 1-cup serving).

## Warm Seasoned Dressing:

1/4 cup plus 3 tablespoons canned low-sodium chicken broth, undiluted

Remove stems from spinach; discard stems. Wash leaves thoroughly, and pat dry with paper towels. Set spinach aside.

Combine chutney and next 5 ingredients in container of an electric blender or food processor; top with cover, and process 30 seconds or until combined.

## Use high-quality ingredients

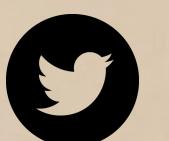
# semantic HTML



## SAVOR THE FLAVOR

When using fresh herbs, keep in mind that fresh herbs are milder in flavor than dried herbs. As a rule of thumb, add three times the amount of a fresh herb to a recipe to match the intensity of flavor of the dried variety. For instance, if a recipe calls for 1/4 teaspoon dried whole thyme,

# FRESH SPRINGTIME SALAD



@resource11

1 pound fresh asparagus spears  
1 cup torn fresh spinach  
2 cups torn red leaf lettuce  
2 cups torn leaf lettuce  
2 cups torn leaf lettuce  
Warm Seasoned Dressing

Snap off tough ends of asparagus with a knife or vegetable peeler. Cut asparagus into 2-inch pieces. Place asparagus in a vegetable steamer and steam over water. Cover and steam 4 to 5 minutes until crisp-tender.

Combine asparagus and next 5 ingredients in a large bowl, tossing gently. Pour dressing over salad; toss gently. Serve immediately. Yield: 8 servings (30 calories per 1-cup serving).

## Warm Seasoned Dressing:

$\frac{1}{4}$  cup plus 3 tablespoons canned low-sodium chicken broth, undiluted

Remove stems from spinach; discard stems. Wash leaves thoroughly, and pat dry with paper towels. Set spinach aside.

Combine chutney and next 5 ingredients in container of an electric blender or food processor; top with cover, and process 30 seconds or until combined.

Mix in seasonings

**ARIA ...just a touch**



### SAVOR THE FLAVOR

When using fresh herbs, keep in mind that fresh herbs are milder in flavor than dried herbs. As a rule of thumb, add three times the amount of a fresh herb to a recipe to match the intensity of flavor of the dried variety. For instance, if a recipe calls for  $\frac{1}{2}$  teaspoon dried whole thyme, add 1 1/2 teaspoons fresh thyme.

# FRESH SPRINGTIME SALAD



1 pound fresh asparagus spears  
1 cup torn fresh spinach  
2 cups torn red leaf lettuce  
2 cups torn leaf lettuce  
Warm Seasoned Dressing  
- off tough ends of asparagus. Do...

Remove stems from spinach; discard stems. Wash leaves thoroughly, and pat dry with paper towels. Set spinach aside.

Combine chutney and next 5 ingredients in container of an electric blender or food processor; top with ... until ... , and process 30 seconds or ... , scraping sides of container.

orange sections, in a large bowl. salad; toss gently. servings (90 calories

E 14.9 / CHOLESTEROL 0 /

Follow the directions

# documentation

large bowl, tossing gently. Pour dressing over salad; toss gently. Serve immediately. Yield: 9 servings (30 calories per 1-cup serving).

## Warm Seasoned Dressing:

$\frac{1}{4}$  cup plus 3 tablespoons canned low-sodium chicken broth, undiluted



### SAVOR THE FLAVOR

When using fresh herbs, keep in mind that fresh herbs are milder in flavor than dried herbs. As a rule of thumb, add three times the amount of a fresh herb to a recipe to match the intensity of flavor of the dried variety. For instance, if dried whole thyme

# FRESH SPRINGTIME SALAD



@resource11

1 pound fresh asparagus spears  
1 cup torn fresh spinach  
2 cups torn red leaf lettuce  
2 cups torn green leaf lettuce

Read helpful hints

## best practices

crisp-tender.

Combine asparagus and next 3 ingredients in a large bowl, tossing gently. Pour dressing over salad; toss gently. Serve immediately. Yield: 9 servings (30 calories per 1-cup serving).

### Warm Seasoned Dressing:

$\frac{1}{4}$  cup plus 3 tablespoons canned low-sodium chicken broth, undiluted

Remove stems from spinach; discard stems. Wash leaves thoroughly, and pat dry with paper towels. Set spinach aside.

Combine chutney and next 5 ingredients in container of

blender or food processor 30 seconds or sides of container.

Add apple, orange sections,monds in a large bowl. over salad; toss gently. Yield: 8 servings (90 calories

FAT 3.5 / CARBOHYDRATE 14.9 / CHOLESTEROL 0 /  
CALORIES 101 / CALCIUM 37



### SAVOR THE FLAVOR

When using fresh herbs, keep in mind that fresh herbs are milder in flavor than dried herbs. As a rule of thumb, add three times the amount of a fresh herb to a recipe to match the intensity of flavor of the dried variety. For instance, if



# Icons

 checkmark-box

 checkmark-circle

 checkmark

 chevron-down

 chevron-left

 chevron-right

 chevron-up

 close-x

 code

 collection



# Icons

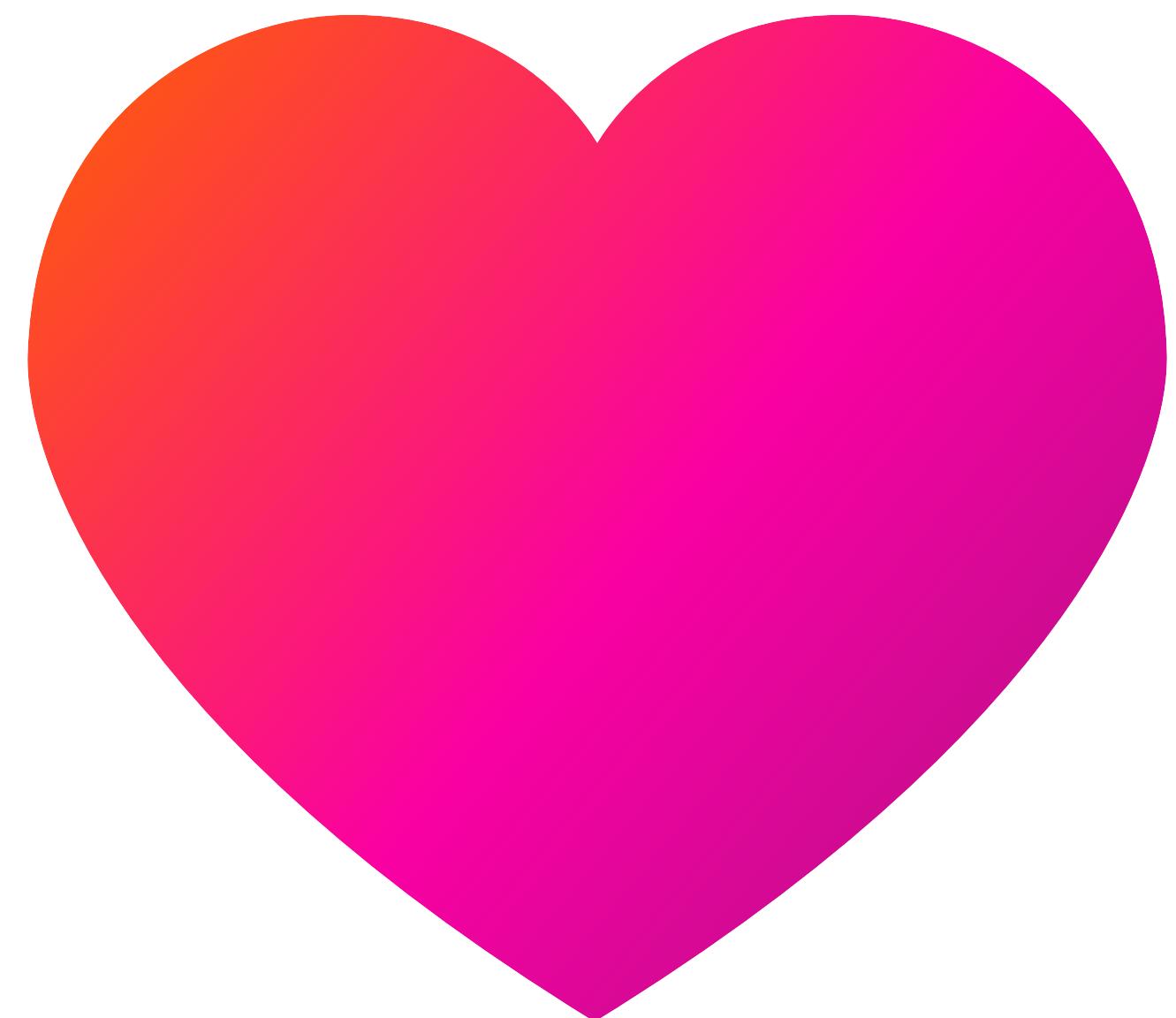
Informative or decorative

Informative icons  
need descriptive text

Decorative icons  
need to be hidden  
from assistive  
technology



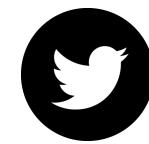
# What does an icon mean, anyway?



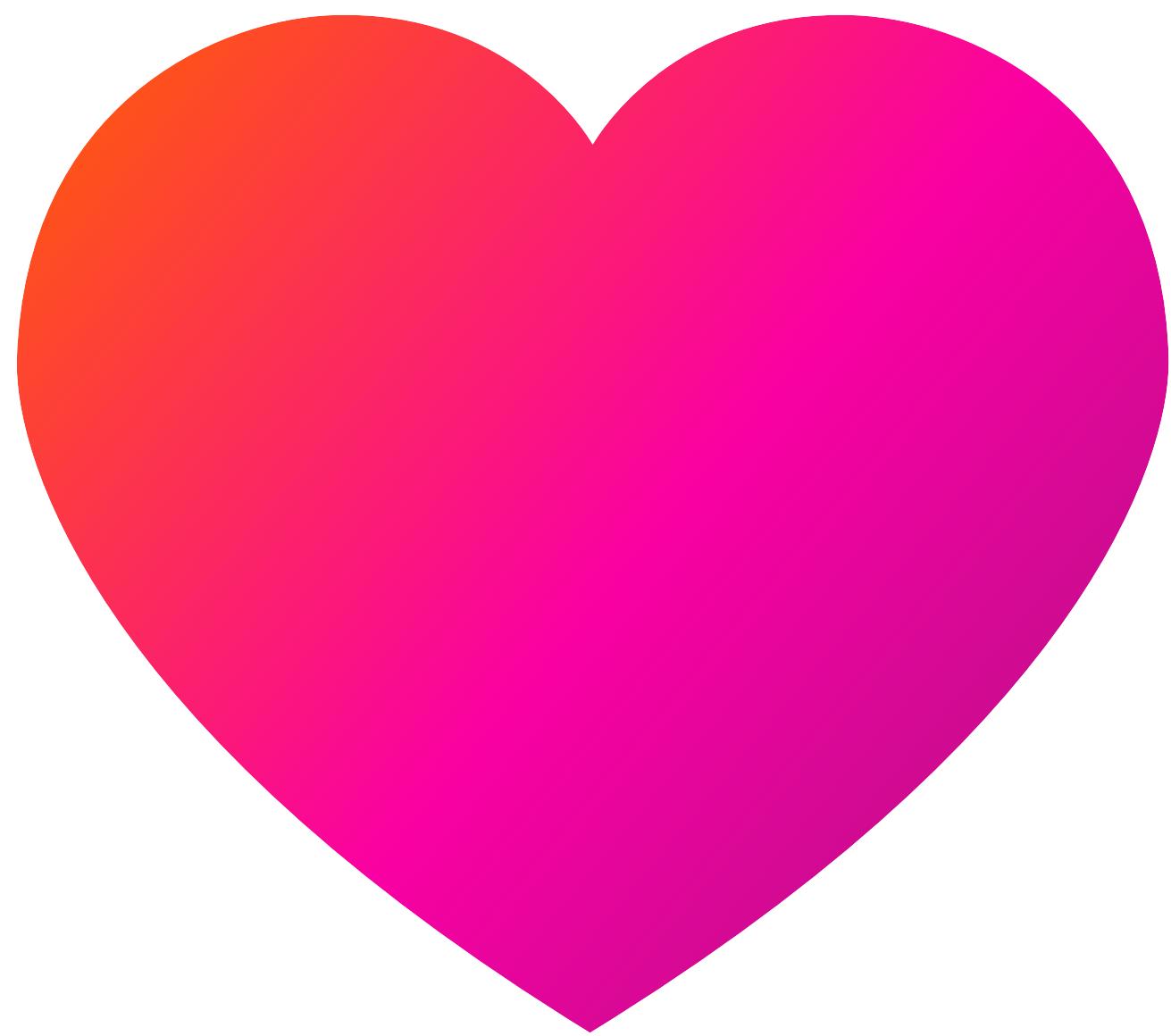




**Pair icons with text  
...when possible**



@resource11



**Favorite**



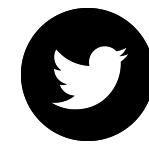
**Favorite**



# SVGs vs. Icon Fonts

VoiceOver  
bugs make  
SVG sadness





# Icon fonts to the rescue



---

# Icon font pattern

```
<span class="icon icon-email"></span>
```



# Accessible icon pattern

```
<span class="root">  
  <span class="icon icon-email"  
        aria-hidden="true"></span>  
  <span class="visuallyHidden">email</span>  
</span>
```

# Icon font

```
<span class="root" style="color: red; font-size: 100px; font-family: 'ORM Icons';">
```

```
.icon {  
    color: currentColor;  
    font-family: "ORM Icons";  
}
```

```
<span class="icon icon-email" style="font-size: 1em; color: green; font-family: 'ORM Icons';">
```

```
    aria-hidden="true">><.icon-email::before {  
        content: "\f12f";  
    }</span>
```



# Hide icon font from screen readers

```
<span class="root">  
  <span class="icon icon-email">  
    aria-hidden="true"><span class="visuallyHidden">  
      .icon-email::before {  
        content: "\f12f";  
      }</span>
```



# Descriptive icon name

```
<span class="root">  
  <span class="icon icon-email"  
        aria-hidden="true"></span>  
    <span class="visuallyHidden">email</span>  
</span>
```

# Visually-hidden text

```
<span class="root">  
  <span class="icon icon-email"  
        aria-hidden="true"></span>  
  <span class="visuallyHidden">email</span>  
</span>
```

```
.visuallyHidden {  
  position: absolute;  
  overflow: hidden;  
  clip: rect(0 0 0 0);  
  height: 1px;  
  width: 1px;  
  margin: -1px;  
  padding: 0;  
  border: 0;  
}
```



# Margin/padding support

```
<span class="root">  
  <span class="icon icon-email"  
        aria-hidden="true"></span>  
  
  <span class="visuallyHidden">email</span>  
</span>
```

```
.root {  
  display: inline-block;  
}
```



# Spans all the way down

```
<span class="root">  
  <span class="icon icon-email"  
        aria-hidden="true"></span>  
  <span class="visuallyHidden">email</span>  
</span>
```



---

# Informative or decorative?

```
<span class="root">  
  <span class="icon icon-email"  
        aria-hidden="true"></span>  
  <span class="visuallyHidden">email</span>  
</span>
```



---

# Informative icons should announce

```
<span class="root">  
  <span class="icon icon-email"  
        aria-hidden="true"></span>  
  <span class="visuallyHidden">email</span>  
</span>
```



# Hide decorative icons from screen readers

```
<span class="root" aria-hidden="true">  
  <span class="icon icon-email"  
        aria-hidden="true"></span>  
  
  <span class="visuallyHidden">email</span>  
  
</span>
```



# React functional component

```
export const Icon = props => {  
  return (  
    <span class="root">  
      <span class="icon icon-email" aria-hidden="true" />  
      <span class="visuallyHidden">email</span>  
    </span>  
  );  
}  
  
export default Icon;
```

## HTML syntax



```
export const Icon = props => {
  class => className
  return (
    <span className="root">
      <span className="icon icon-email" aria-hidden={true} />
      <span className="visuallyHidden">email</span>
    </span>
  );
}
```

```
export default Icon;
```

**class => className**

**“true” => {true}**

**aria-hidden={true} />**

**Close empty span**



# JSX syntax

```
export const Icon = props => {  
  
  return (  
    <span className="root">  
      <span className="icon icon-email" aria-hidden={true} />  
      <span className="visuallyHidden">email</span>  
    </span>  
  );  
}  
  
export default Icon;
```



```
export const Icon = props => {
  const { iconHidden, iconName, iconTitle } = this.props;
  const icon = Icons[iconName];
  if (!icon) return null;

  return (
    <span className="root" aria-hidden={iconHidden ? true : null}>
      <span
        className={`icon icon-${iconName}`}
        aria-hidden={true}
      />
      <span className="visuallyHidden">
        {iconTitle || iconName}
      </span>
    </span>
  );
}

export default Icon;
```

Support incoming  
props, add guardrails



@resource11

# Guardrails





```
export const Icon = props => {
  const { iconHidden, iconName, iconTitle } = this.props;
  const icon = Icons[iconName];
  if (!icon) return null;

  return (
    <span className="root" aria-hidden={iconHidden ? true : null}>
      <span
        className={`icon icon-${iconName}`} aria-hidden={true}
      />
      <span className="visuallyHidden">
        {iconTitle || iconName}
      </span>
    </span>
  );
}

export default Icon;
```

## Icon naming guardrails



```
export const Icon = props => {
  const { iconHidden, iconName, iconTitle } = this.props;
  const icon = Icons[iconName];
  if (!icon) return null;

  return (
    <span className="root" aria-hidden={iconHidden ? true : null}>
      <span
        className={`icon icon-${iconName}`}
        aria-hidden={true}
      />
      <span className="visuallyHidden">
        {iconTitle || iconName}
      </span>
    </span>
  );
}

export default Icon;
```

Icon  
description  
guardrails



```
export const Icon = props => {
  const { iconHidden, iconName, iconTitle } = this.props;
  const icon = Icons[iconName];
  if (!icon) return null;

  return (
    <span className="root" aria-hidden={iconHidden ? true : null}>
      <span
        className={`icon icon-${iconName}`}
        aria-hidden={true}
      />
      <span className="visuallyHidden">
        {iconTitle || iconName}
      </span>
    </span>
  );
}

export default Icon;
```

## Icon Hiding guardrails



```
export const Icon = props => {
  const { iconHidden, iconName, iconTitle } = this.props;
  const icon = Icons[iconName];
  if (!icon) return null;

  return (
    <span className="root" aria-hidden={iconHidden ? true : null}>
      <span
        className={`icon icon-${iconName}`}
        aria-hidden={true}
      />
      <span className="visuallyHidden">
        {iconTitle || iconName}
      </span>
    </span>
  );
}

export default Icon;
```

## Icon Hiding guardrails



...but SVGs!

6

## Now Brewing... Font Awesome 6!

The next generation of our icon library + toolkit is coming with [more icons](#), [more styles](#), [more services](#), and [more awesome](#). Pre-order now to get early access and releases over the next year!

[Learn More!](#)

# Font Awesome



Get vector icons and social logos on your website with Font Awesome, the web's most popular icon set and toolkit.

[Start for Free](#)[Get More with Pro](#)



## Now Brewing... Font Awesome 6!

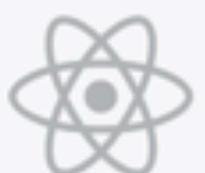
The next generation of our icon library + toolkit is coming with [more icons](#), [more styles](#), [more services](#), and [more awesome](#). Pre-order now to get early access and releases over the next year!

[Learn More!](#)

On the Web

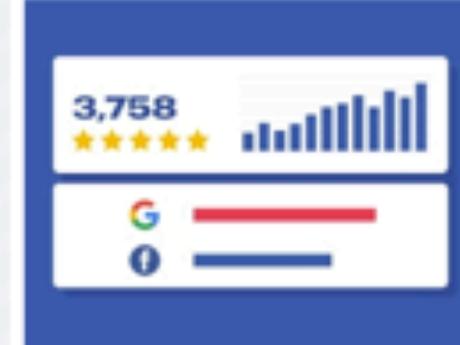
On the Desktop

Version: 5.14.0 ▾



## React

Font Awesome now has an official React component that's available for a friction-less way to use our icons in your React applications.



It's time to take control of your online reputation. Get more reviews on the sites that matter most, for free.

ads via Carbon



Before we start, review the following and make sure you're...

- wanting to use React and not React Native - that's a different [react-native-fontawesome](#) component.
- okay with using SVG + JS to render icons in your project - this react component [relies on that technology](#).
- already [set up to use Pro packages](#) if you plan on using Pro-only icons.

### Set Up

[Getting Started](#)

[Upgrading from Version 4](#)

[Hosting Font Awesome Yourself](#)

[Using a Package Manager](#)

# Font Awesome works how and where you need it to



Web Fonts + CSS



SVG + JS



Sketch



Adobe Apps



Vue.js



Angular



React



ember



**Accessibility-Minded**  
Font Awesome loves screen readers! With our auto-accessibility feature, it's super simple to do the right thing for your users.



**Desktop-Friendly**  
Font Awesome is fantastic to work with on the desktop, especially with all-new ligatures. Try it in your next design or



**Tried & Tested**  
We've stress-tested them, so that your icons and styling work and display perfectly in all modern browsers.



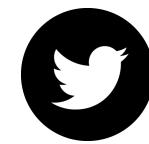
**Pixel-Perfect Rendering**  
Our icons are meticulously designed on a grid that helps them render perfectly and makes them legible at



```
<FontAwesomeIcon  
icon="coffee"  
/>
```



```
<svg  
    data-icon="coffee" aria-hidden="true"  
    focusable="false"  
/>  
  
<path>...</path>  
  
</svg>
```



```
<FontAwesomeIcon
```

```
    icon="coffee"
```

```
    ariaLabel="beverage"
```

/>



```
<FontAwesomeIcon
```

```
  icon="coffee"
```

```
    ariaLabel="beverage"
```

/>



```
<svg
```

```
  data-icon="coffee" aria-hidden="true"
```

```
  focusable="false" aria-label="beverage"
```

```
/>
```

```
<path>...</path>
```

```
</svg>
```



<svg

# aria-hidden hides from screen reader

/>

<path>...</path>  
</svg>

data-icon="coffee"

**aria-hidden="true"**

focusable="false"

**aria-label="beverage"**

aria-label will  
not announce



```
export const Icon = props => {
  const { iconHidden, iconName, iconTitle } = this.props;

  return (
    <span className="root" aria-hidden={iconHidden ? true : null}>
      <span
        className={`icon icon-${iconName}`}
        aria-hidden={true}
      />
      <span className="visuallyHidden">
        {iconTitle || iconName}
      </span>
    </span>
  );
}

export default Icon;
```

**Swap out font-based  
className**



```
export const Icon = props => {
  const { iconHidden, iconName, iconTitle } = this.props;

  return (
    <span className="root" aria-hidden={iconHidden ? true : null}>
      <FontAwesomeIcon
        icon={iconName}
      />
      <span className="visuallyHidden">
        {iconTitle || iconName}
      </span>
    </span>
  );
}

export default Icon;
```

Swap in FontAwesomelcon



```
export const Icon = props => {
  const { iconHidden, iconName, iconTitle } = this.props;

  return (
    <span className="root" aria-hidden={iconHidden ? true : null}>
      <FontAwesomeIcon
        icon={iconName}
      />
      <span className="visuallyHidden">
        {iconTitle || iconName}
      </span>
    </span>
  );
}

export default Icon;
```

**Informative icon  
naming supported**



# Buttons



Basic Button

Basic Button Default

Basic Button

Basic Button Hover

Featured Button

Featured Button Default

Featured Button

Featured Button Hover/Focus

Featured 

Featured Button w/ Icon Default

Featured 

Featured Button w/ Icon Hover/Focus

Disabled Button

Disabled Button Default



# Buttons

Perform an action on  
the page

Should look and act  
like a button

Screen reader and  
keyboard  
functionality for free



---

# Button

```
<button class="root">
```

Send an email

```
</button>
```



---

# Button

Sprinkle  
some ARIA

```
<button class="root" aria-label="Read more about Dinosaurs">  
  Read more  
</button>
```



# Button

Sprinkle  
some ARIA

```
<button class="root" aria-label="Read more about dinosaurs">  
  Read more</button>
```

Button text and aria label  
start with same words



# Button

Sprinkle  
some ARIA

```
<button class="root" aria-label="Read more about dinosaurs">  
  Read more</button>
```

Button text and aria label  
start with same words



---

# Button

```
<button className="root" aria-label="Read about Dinosaurs">  
  Read More  
</button>
```



```
<button className="root" aria-label="Read about Dinosaurs>  
  {children}  
    <Icon  
      iconName={iconName}  
      iconHidden={true}  
    />  
</button>
```

Add icon  
support



```
<button className="root" aria-label="Read about Dinosaurs  
    onClick={onClick} disabled={disabled ? true : null}>  
      
        {children}  
        <Icon  
            iconName={iconName}  
            iconHidden={true} />  
    </span>  
</button>
```

Wrap contents  
in span



```
export const Button = props => {
  const { ariaLabel, children, disabled, iconName, onClick } = this.props;

  return (
    <button className="root" aria-label={ariaLabel}>
      <onClick={onClick} disabled={disabled ? true : null}>
        <span className="btnContentWrap">
          {children}
          {iconName && (
            <Icon
              iconName={iconName}
              iconHidden={true}
            />
          )}
        </span>
      </button>
  );
}
```

Add props, click  
handler &  
disabled button  
support



```
export const Button = props => {
  const { ariaLabel, children, disabled, iconName, onClick } = this.props;

  return (
    <button className="root" aria-label={ariaLabel}
      onClick={onClick} disabled={disabled ? true : null}>
      <span className="btnContentWrap">
        {children}
        {iconName && (
          <Icon
            iconName={iconName}
            iconHidden={true}
          />
        )}
      </span>
    </button>
  );
}
```

## Add guardrails

No iconName?  
No icon rendered



# Inputs

Label

Input Default - Medium

Input Label

Some text here

Input Focused - Medium

Input Label

Some text here

⚠ Error text

Input Invalid - Medium

Label

Input Disabled - Medium

👤 Label

Input with Icon, Default - Medium

Label

👤 Some text here

Input with Icon, Focused - Medium

Label

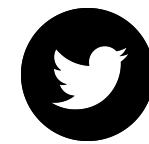
👤 Some text here

⚠ Error text

Input with Icon, Invalid - Medium

👤 Label

Input with Icon, Disabled - Medium



# Inputs need labels and error messages

Labeled inputs  
give all users  
more context



---

# Placeholders are NOT labels

Avoid using placeholders instead of labels, users will lose context

Hard to style across browsers

Placeholders aren't auto translated



# Avoid Horizontal Scrolling

Max input width:  
80 characters

Keep labels stacked vertically in close proximity

Labels above input,  
errors below input



```
<div className="root">
  {label && (
    <label htmlFor={id} className="InputLabel">{label}</label>
    {iconName && (
      <Icon
        iconName={iconName}
        iconHidden={true}
      />
    )}

    <input
      className={inputClasses}
      id={id}
      name={id}
      type={type}
      value={value}
      disabled={disabled ? true : null}
      onChange={onChange}
      onKeyPress={onKeyPress}
      aria-describedby={invalid && error ? `error-${id}` : null}
      aria-invalid={invalid ? true : null}
      aria-required={required ? true : null}
    />
  )
  {invalid &&
    error &&
      <div className="errorWrapper">
        <Icon iconName="warning-fill"
          iconHidden={true} />
        <span
          id={`error-${id}`}
          className="errorTxt"
          aria-live="polite"
        >
          {error}
        </span>
      </div>
  }
</div>
```



```
<div className="root">
  {label && (
    <label htmlFor={id} className="InputLabel">{label}</label>
    {iconName && (
      <Icon
        iconName={iconName}
        iconHidden={true}
      />
    )}
    <input
      className={inputClasses}
      id={id}
      name={id}
      type={type}
      value={value}
      disabled={disabled ? true : null}
      onChange={onChange}
      onKeyPress={onKeyPress}
      aria-describedby={invalid && error ? `error-${id}` : null}
      aria-invalid={invalid ? true : false}
      aria-required={required ? true : null}
    />
  )
  {invalid && error && (
    <div className="errorWrapper">
      <Icon iconName="warning-filled" id="errorIcon" />
      <span
        id="errorText"
        className="errorTxt"
        aria-live="polite"
      >{error}</span>
    </div>
  )
}
```

# Semantic HTML



```
<div className="root">
  {label && (
    <label htmlFor={id} className="InputLabel">{label}</label>
    {iconName && (
      <Icon
        iconName={iconName}
        iconHidden={true}
      />
    )}
    <input
      className={inputClasses}
      id={id}
      name={id}
      type={type}
      value={value}
      disabled={disabled ? true : null}
      onChange={onChange}
      onKeyPress={onKeyPress}
      aria-describedby={invalid && error ? `error-${id}` : null}
      aria-invalid={invalid ? true : null}
      aria-required={required ? true : null}
    />
  )
  {invalid &&
  error && (
    <div className="errorWrapper">
      <Icon iconName="warning-filled" id="errorIcon" />
      <span id="errorFoo" class="errorTxt">
        Whoops! Error.
      </span>
    </div>
  )
}
</div>
```

# Labels and error messages



```
<div className="root">
  {label && (
    <label htmlFor={id} className="InputLabel">{label}</label>
    {iconName && (
      <Icon
        iconName={iconName}
        iconHidden={true}
      />
    )}
    <input
      className={inputClasses}
      id={id}
      name={id}
      type={type}
      value={value}
      disabled={disabled ? true : null}
      onChange={onChange}
      onKeyPress={onKeyPress}
      aria-describedby={invalid && error ? `error-${id}` : null}
      aria-invalid={invalid ? true : false}
      aria-required={required ? true : null}
    />
  )
  {invalid && error && (
    <div className="errorWrapper">
      <Icon iconName="warning-filled" id="errorIcon" />
      <span id="errorFoo" className="errorTxt">
        Whoops! Error.
      </span>
    </div>
  )
}
```

Associate  
label with  
input



```
<div className="root">
  {label && (
    <label htmlFor={id} className="InputLabel">{label}</label>
  )}
  {iconName && (
    <Icon
      iconName={iconName}
      iconHidden={true}
    />
  )}
  <input
    className={inputClasses}
    id={id}
    name={id}
    type={type}
    value={value}
    disabled={disabled ? true : null}
    onChange={onChange}
    onKeyPress={onKeyPress}
    aria-describedby="errorFoo"
    aria-invalid={invalid ? true : null}
    aria-required={required ? true : null}
  />
  {invalid && error && (
    <div className="errorWrapper">
      <Icon iconName="warning-fill" />
      <span id="errorFoo" className="errorTxt">
        Whoops! Error.
      </span>
    </div>
  )}
</div>
}
```

# Sprinkle some ARIA



```
<div className="root">
  {label && (
    <label htmlFor={id} className="InputLabel">
      {iconName && (
        <Icon
          iconName={iconName}
          iconHidden={true}
        />
      )}
      <input
        className={inputClasses}
        id={id}
        name={id}
        type={type}
        value={value}
        disabled={disabled ? true : null}
        onChange={onChange}
        onKeyPress={onKeyPress}
        aria-describedby={invalid & error ? `error-${id}` : null}
        aria-invalid={invalid ? true : null}
        aria-required={required ? true : null}
      />
    )
  }
  {invalid &&
    error && (
      <div className="errorWrapper">
        <Icon iconName="warning-fill" />
        <span id="errorFoo" className="errorTxt" aria-live="polite">
          Whoops! Error.
        </span>
      </div>
    )
  }
</div>
```



```
<div className="root">
  <label htmlFor={id} className="InputLabel">
    {label}
<div className="root">
  {label && (
    <label htmlFor={id} className="InputLabel">{label}/>
    {iconName && (
      <Icon
        iconName={iconName}
        iconHidden={true}
        id={id}
      />
    )}>
    <input
      className={inputClasses}
      id={id}
      name={id}
      type={type}
      value={value}
      disabled={disabled ? true : null}
      onChange={onChange}
      onKeyPress={onKeyPress}
      aria-describedby={invalid && error ? `error-${id}` : null}
      aria-invalid={invalid ? true : null}
      aria-required={required ? true : null}
    />
  )}>
  {invalid && error && (
    <span className="errorWrapper">
      <Icon iconName="warning-fill"
        iconHidden={true}
        id={`error-${id}`}
      />
      <span
        id={`error-${id}`}
        className="errorTxt"
        aria-live="polite"
        aria-live="polite"
      >
        {error} {error}
      </span>
    </span>
  )}>
</div>
```

# disabled and synthetic event support

and true/null handling



```
{label && (
  <label htmlFor={id} className="InputLabel">
    {label}
  </label>
  {label && (
    <label htmlFor={id} className="InputLabel">{label}</label>
    {iconName && (
      <Icon
        iconName={iconName}
        iconHidden={true}
        title={iconTitle}
      />
    ) }
    <input
      className={inputClasses}
      id={id}
      aria={id}
      type={type}
      value={value}
      disabled={disabled ? true : null}
      onChange={onChange}
      onKeyPress={onKeyPress}
      aria-describedby={invalid && error ? `error-${id}` : null}
      aria-invalid={invalid ? true : null}
      aria-required={required ? true : null}
    type={type}
    value={value}
    disabled={disabled ? true : null}
    onChange={onChange}
    onKeyPress={onKeyPress}
    aria-describedby={invalid && error ? `error-${id}` : null}
    aria-invalid={invalid ? true : null}
    aria-required={required ? true : null}
  ) }
)
```

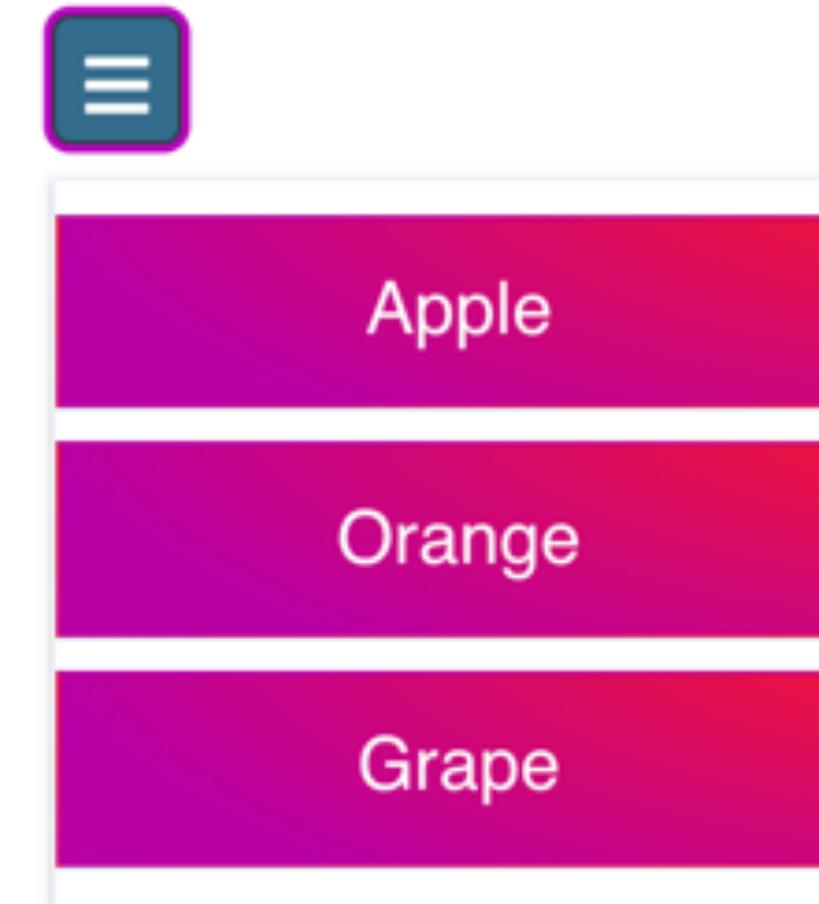
# Add guardrails



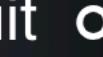
```
<div className="root">
  {label && (
    <label htmlFor={id} className="InputLabel">{label}</label>
    {iconName && (
      <Icon
        iconName={iconName}
        iconHidden={true}
      />
    )}
    <input
      className={inputClasses}
      id={id}
      name={id}
      type={type}
      value={value}
      disabled={disabled ? true : null}
      onChange={onChange}
      onKeyPress={onKeyPress}
      aria-describedby={invalid && error ? `error-${id}` : null}
      aria-invalid={invalid ? true : null}
      aria-required={required ? true : null}
    />
  )} {invalid && error && (
    <div className="errorWrapper">
      <Icon iconName="warning-fill"
            iconHidden={true} />
      <span
        id={`error-${id}`}
        className="errorTxt"
        aria-live="polite"
      >
        {error}
      </span>
    </div>
  )} {error}
</span>
</div>
</div>
}
```

# Add guardrails

# Disclosure Widgets



Hungry for more interesting fruit facts?  
View a [full list of fruit trivia](#) in at our  
virtual farm stand.

Find a fruit 

Search

Go



# Disclosure Widgets

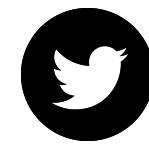
Flexible pattern

Keyboard and  
mouse operable

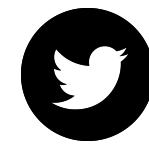
Shared base  
functionality



# Menus & Toggle Tips



**Tooltips are NOT  
Toggle Tips**

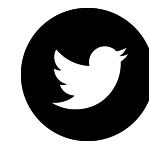


# Tooltips can be problematic



# Primary keyboard interactions

Space bar/Enter key  
Escape key  
Mouse clicks



# **Targeted focus management**

# Disclosure Widgets Demo



Apple

Orange

Grape



Hungry for more interesting fruit facts?

View a [full list of fruit trivia](#) in at our  
virtual farm stand.

Find a fruit 

Search

Go



Document,  
document,  
document

TRIED AND TRUE RECIPES

Fresh Pineapple Ring

Waxed Grill Dinner

Slices

Grilled Tomatoes

Grilled Pineapple

Fresh Vegetable Salad

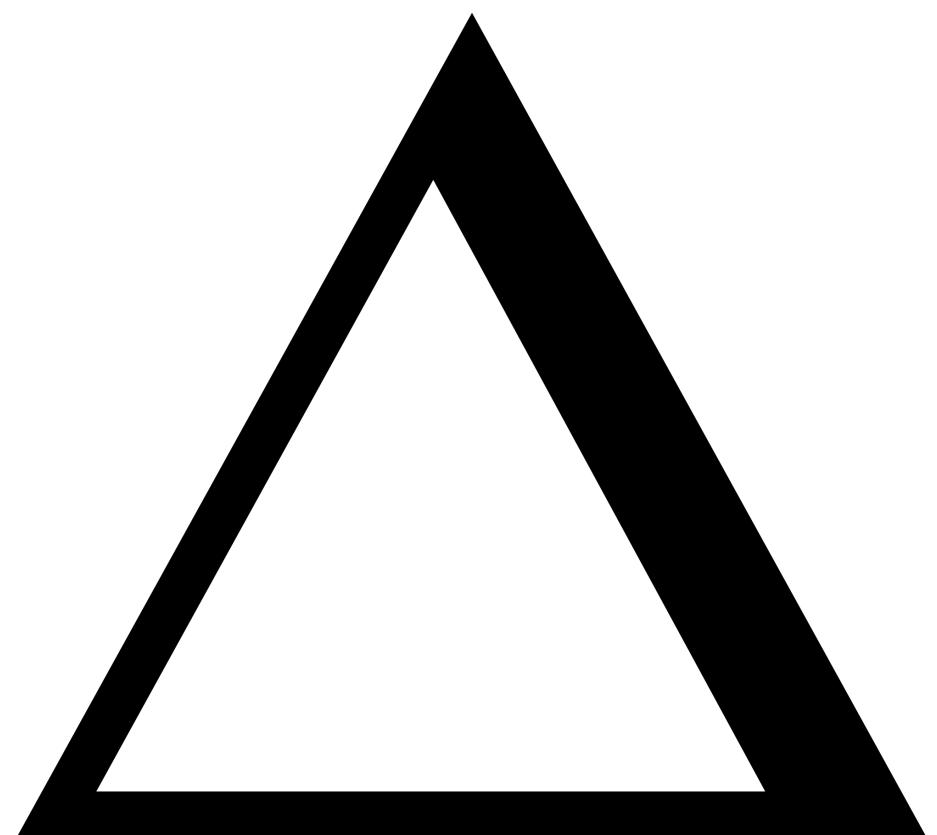
Coupe Esperance

Daffodil Cake



+









@resource11



# Storybook



Press "/" to search...

#### EXPERIMENTAL

- ▶ Avatar
- ▶ BlockContainer
- ▶ CardContainer
- ▶ Experimental CardGroup Patterns
- ▶ Carousel
- ▶ Tag
- ▶ Toggle

#### CORE

- ▶ Button
- ▶ ByLine
- ▶ CallToAction
- ▶ Card
  - Default using max props, div wrapper, truncated text

**Shorter body copy has content tolerance, except for in IE11**

max props using CardGroup wrapper

full-width horizontal card in CardGroup

using columnSizes for all

using mixed google.com immSizes



#### LEARNING PATH

OCTOBER 26, 1985

essay because it is too long to show the entire thing

Madsen, Martín Bravo, Kathleen McMahon, Pauline Masigla, S. Kimball, James Touhey and Paul

This text describes the contents of the article illustrated in the image above donec consectetur...

foo



#### LEARNING PATH

OCTOBER 26, 1985

essay because it is too long to show the entire thing

Madsen, Martín Bravo, Kathleen McMahon, Pauline Masigla, S. Kimball, James Touhey and Paul

This text describes the contents of the article.

foo



#### LEARNING PATH

OCTOBER 26, 1985

A super short title

By Kathleen McMahon, Pauline Masigla, S. Kimball and James Touhey

This text describes the contents of the article.

foo



1 Violations

10 Passes

1 Incomplete

Highlight results

Ensures buttons have discernible text

Buttons must have discernible text

More info...

1. article:nth-child(1) > .orm-Card-contentWrapper > footer > div > .orm-Button-root.orm-Button-small.orm-Button-secondary

#### Critical

Element does not have inner text that is visible to screen readers

#### Serious

aria-label attribute does not exist or is empty

#### Serious

aria-labelledby attribute does not exist, references elements that do not exist or references elements that are empty

#### Minor

Element's default semantics were not overridden with role="presentation"

#### Minor

Element's default semantics were not overridden with role="none"

#### Serious

Element has no title attribute  
title attribute is empty

Rerun tests



# Helpful hints

## Informative Icon

### Custom icon name

Each icon has a visually-hidden `title` describing it's meaning, and that `title` is announced to screen readers by default. If you want the icon to be announced with a **different name** to screen readers, pass in a different string to the `title` prop.



```
<Icon name="search" title="monocle on a stick" />
```



# Helpful hints

## Decorative Icon

### Hidden from screen readers

Each icon has a visually-hidden title describing its meaning and that title is announced to screen readers by default. If you don't want the title announced to screen readers, pass in `iconHidden={true}` to the component. Otherwise, simply omit the prop.



```
<Icon name="search" iconHidden={true} />
```



# Helpful hints

## Prop tables

Name	Type	Description
color	enum	Icon color.
extraClasses	theme	A map of CSS classes to be appended to DOM nodes internally. Valid keys are: See <a href="#">theme</a> .
iconHidden	bool	Used to prevent screen readers from announcing the icon. Best practice is to set iconHidden to true or omit the prop.
name	enum	<b>Required.</b> Icon name.
onClick	func	Called when this icon is clicked.
size	responsive	Icon size. Receives a value matching <code>PropTypes.oneOf(IconSizes)</code> or an object with values assigned to any of these keys: <code>xsmall</code> , <code>xsmall1600</code> , <code>small</code> , <code>small1960</code> , <code>medium</code> . See <a href="#">responsive</a> .
title	string	Used to override the default visually-hidden Icon title that is announced by screen readers.



# Helpful hints

## Component dos/don'ts



**Do:**

- Use a Basic button for a "safe" user action such as 'Cancel'.
- Use Button with Icon variant either for decorative purposes or to provide extra information to the user.
- Pass the text as your Button's children, unless you're using an icon-only button.
- For icon-only buttons, pass the text through your Button's ariaLabel prop.
- Start your ariaLabel text with same text as your Button's children text when using both props.



**Don't:**

- Use a Destructive button for 'Cancel'.
- Use a Feature Button with icon as a visual indicator the user is leaving the page. Use the Link with icon variant instead.
- Pass in block elements or links as your Button's children. Use valid HTML.
- Use an icon-only button without setting the ariaLabel prop



Dedicate a page  
to accessibility  
resources

TRIED AND TRUE RECIPES

CONTENTS

	PAGE
USING THIS BOOK	9
HELPFUL HINTS	10
TABLE OF EQUIVALENTS	11
PROPORTIONS	12
WEIGHTS AND MEASURES	13
HEAT AND COOKING	16
TIME TABLES	19
SUGAR TEMPERATURES	19
SCALE OF RANGE	20
REFRIGERATION	21
SUCCESSFUL CANNING	25
TABLE SETTING AND SERVICE	27
CATERING PROBLEMS	30
FIRST AID Hints	31
SUGGESTIONS FOR MENU PLANNING	35
BEVERAGES	41
HORS D'OEUVRES AND CANAPES	65
APPETIZERS	67
SOUPS, STEWS, CHOWDERS	73
	79
	86
	98

# Wrap-up

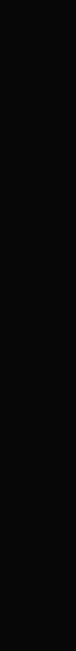
---

# Our users are diverse

Vision



Hearing



Sensory

Language

Low bandwidth



@resource11

Your Design  
system is a  
cookbook

Cook Book



# Cookbooks have personality

**The Personality of a Cookbook**

**The Stable Ingredients**

**The Elusive Charm**

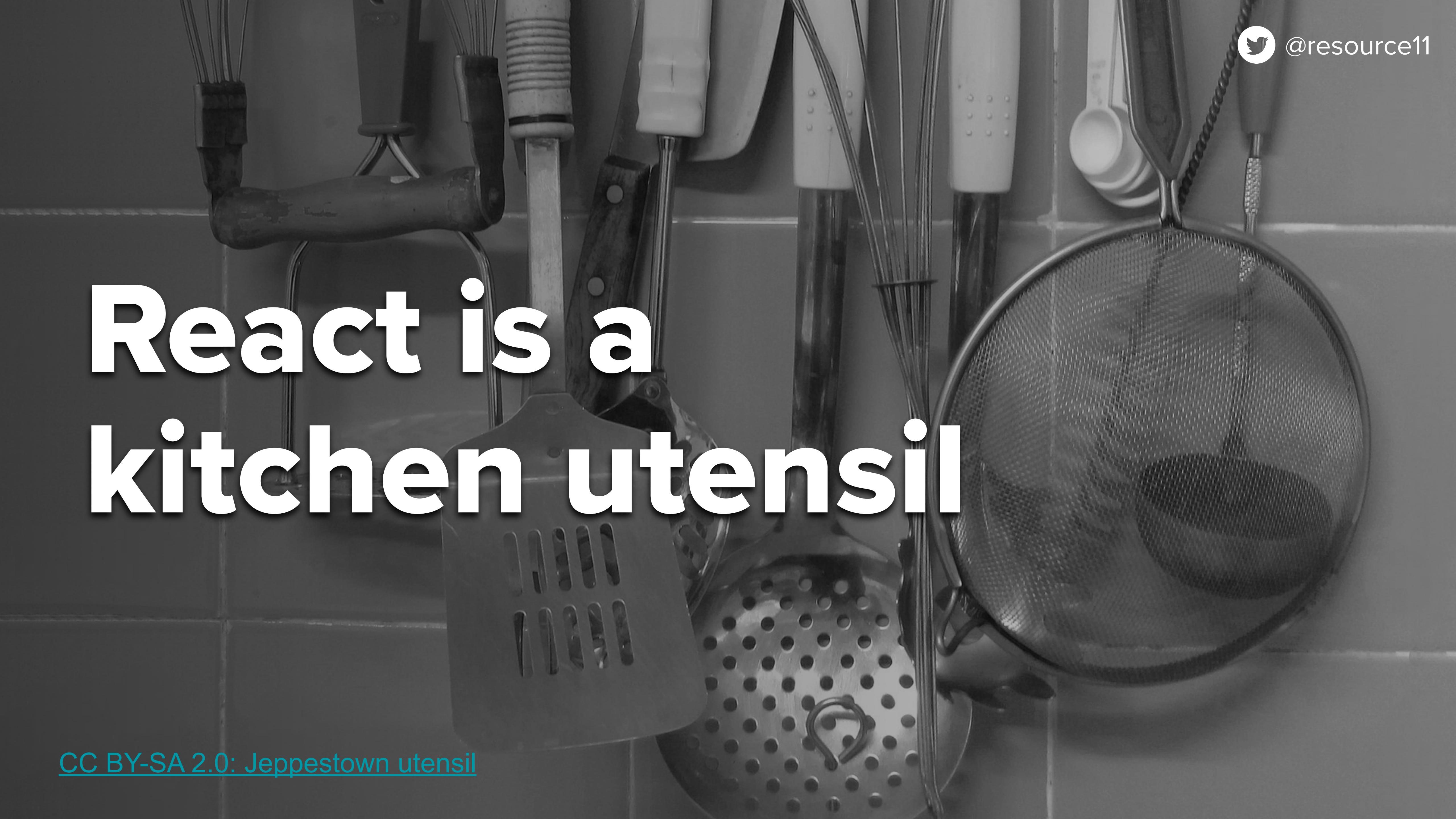
**This Beauty**

posed of known and stable ingredients with unknown and elusive ones to make a mixture as familiar, friendly and exhilarating as a pine woods early on a summer morn.

on preparing and serving foods, based on the needs of the family, an easy handling of all fundamental facts, compounded of a sound knowledge of what is organized, specific, easy-to-follow, step-by-step procedure, a concise, feeling of confidence in the final product.

—a sense of humor—a flair for the clever idea something called style, but above all a feeling for the kind of beauty that comes from clear overtones: a light touch.

has been made possible by the cooperation and confidence food and equipment industry, whose constant flow of inspiring about them in their work-a-day world.



React is a  
kitchen utensil



You are the cook



@resource11

Components are  
your tried and  
true recipes

RUTH WAKEFIELD'S  
TOLL HOUSE

Tried and True Recipes

RUTH GRAVES  
WAKEFIELD  
Dietitian • Lecturer



M. BARROWS & COMPANY, INC.  
New York  
1941



@resource11

# WCAG is your reference material

**new  
revised  
edition**

The image shows the front cover of a book. The title "CAG is Your Encyclopedia Cook Book" is printed in large, bold, white serif capital letters. Behind the title, the words "CULINARY ARTS INSTITUTE" are printed in a smaller, dark serif font. A decorative border at the top features stylized illustrations of various kitchen tools like knives, spoons, and pots.

# materia



Document,  
document,  
document

TRIED AND TRUE RECIPES

Fresh Pineapple Ring  
Baked Grill Dinner  
Slices  
Grilled Tomatoes  
Grilled Pineapple  
Fresh Vegetable Salad  
Coupe Esperance  
Daffodil Cake



@resource11

Dinosaurs are  
**ALWAYS** the  
hotness



# Thank you.

# <https://noti.st/resource11>

Slide deck posted after the talk

# @resource11

[Twitter](#) | [Instagram](#) | [GitHub](#)