debugger;

DENYS MISHUNOV
DIGITAL GARDEN AS
@mishunov

LIVE & BREATHE* CODE

* also drink, eat, and don’t forget to enjoy

```
    return path.slice(0, dotIndex);
  },
  isDeep: function (path) {
    return path.indexOf('.') !== -1;
  },
  isAncestor: function (base, path) {
    return base.indexOf(path + '.') === 0;
  },
  isDescendant: function (base, path) {
    return path.indexOf(base + '.') === 0;
  },
  translate: function (base, newBase, path) {
    return newBase + path.slice(base.length);
  },
  matches: function (base, wildcard, path) {
    return base === path || this.isAncestor(base, path) || Boolean(wildcard) && this.isDescendant(base, path);
  }
};Life.Base._addFeature({
  _prepAnnotations: function () {
    if (!this._template) {
      this._notes = [];
    } else {
      var self = this;
      Life.Annotations.prepElement = function (element) {
        self._prepElement(element);
      };
      if (this._template._content && this._template._content._notes) {
        this._notes = this._template._content._notes;
      } else {
        this._notes = Life.Annotations.parseAnnotations(this._template);
```

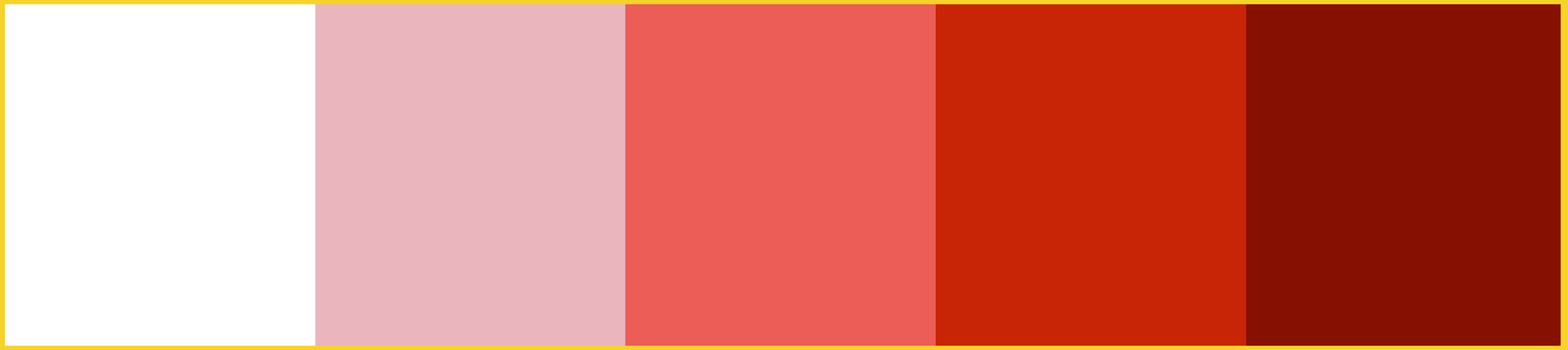
```
    return path.slice(0, dotIndex);
  },
  isDeep: function (path) {
    return path.indexOf('.') !== -1;
  },
  isAncestor: function (base, path) {
    return base.indexOf(path + '.') === 0;
  },
  isDescendant: function (base, path) {
    return path.indexOf(base + '.') === 0;
  },
  translate: function (base, newBase, path) {
    return newBase + path.slice(base.length);
  },
  matches: function (base, wildcard, path) {
    return base === path || this.isAncestor(base, path) || Boolean(wildcard) && this.isDescendant(base, path);
  }
};Life.Base._addFeature({
  prepAnnotations: function () {
```

Elements Network Performance Console 3 1

```
Frustration: repeat count must be less than infinity  life.html:1300
24/7: BRAIN is not a constructor  life.html:1315
Perfectionism: too much recursion  life.html:1316
Narrow-mindedness: unreachable code after return statement  life.html:1325
>
```

```
Frustration: repeat count must be less than infinity
```


DEVEL
OPER



ЗОНЫ ПУЛЬСА


(frustration)

```
    return path.slice(0, dotIndex);
  },
  isDeep: function (path) {
    return path.indexOf('.') !== -1;
  },
  isAncestor: function (base, path) {
    return base.indexOf(path + '.') === 0;
  },
  isDescendant: function (base, path) {
    return path.indexOf(base + '.') === 0;
  },
  translate: function (base, newBase, path) {
    return newBase + path.slice(base.length);
  },
  matches: function (base, wildcard, path) {
    return base === path || this.isAncestor(base, path) || Boolean(wildcard) && this.isDescendant(base, path);
  }
};Life.Base._addFeature({
  prepAnnotations: function () {
```

Elements Network Performance Console

3 1

✕ ▶	Frustration: repeat count must be less than infinity	life.html:1300
✕ ▶	24/7: BRAIN is not a constructor	life.html:1315
✕ ▶	Perfectionism: too much recursion	life.html:1316
⚠ ▶	Narrow-mindedness: unreachable code after return statement	life.html:1325

>

```
✕ ▶ 24/7: BRAIN is not a constructor
```

ЗАКОН
ПАРКИНСОНА

“ ***Работа*** *заполняет*
ровно то ***время***,
что ***на неё отпущено*** ”

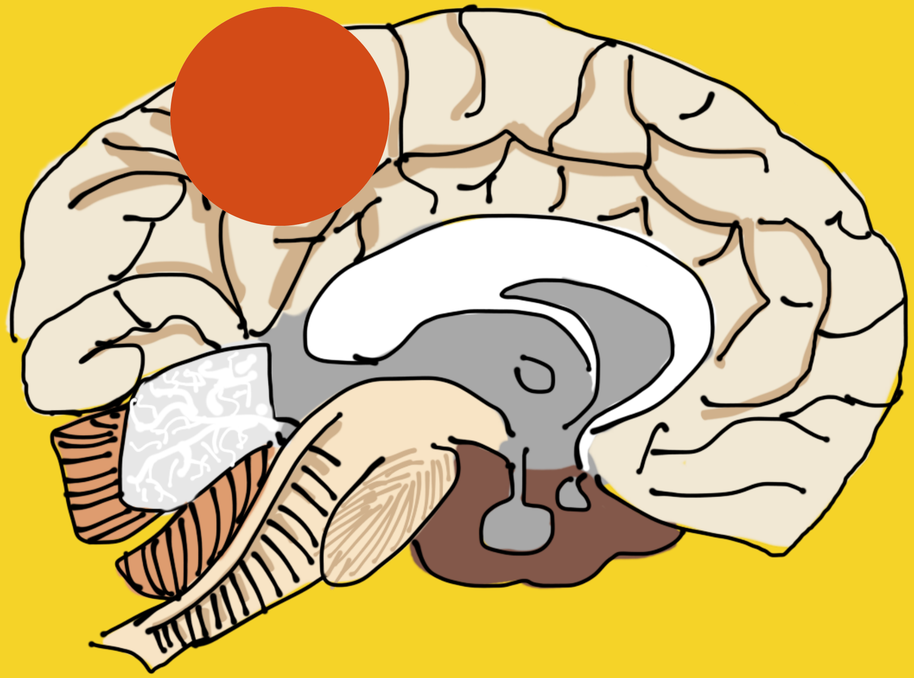
Henry Cooke
@prehensile

Follow

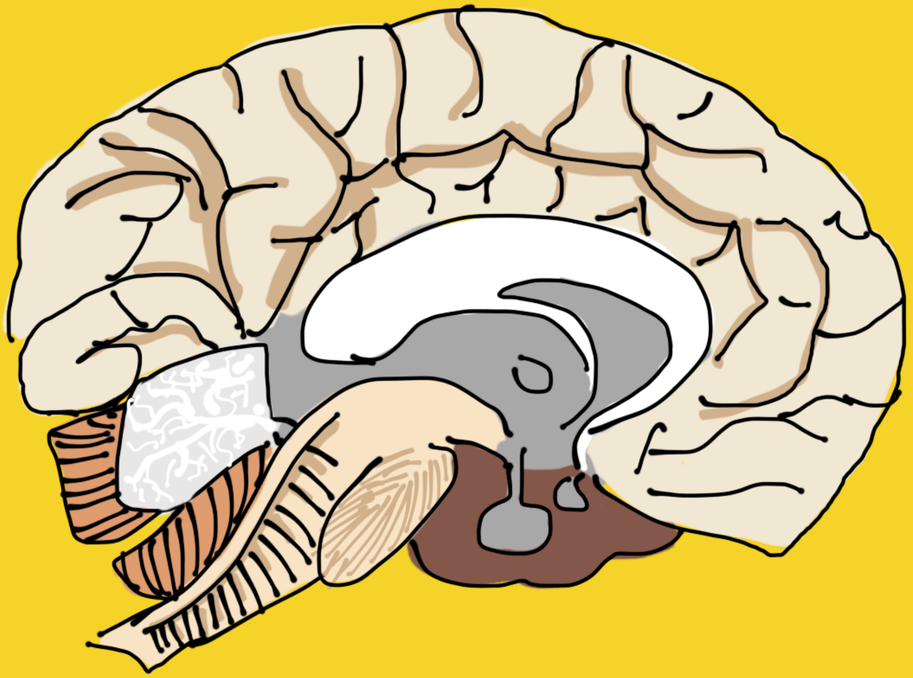
stay up late writing code every day. Everyone knows you write your best stuff after midnight. Sleep is wasted time. #shittyprogrammertips

#йаусталъ

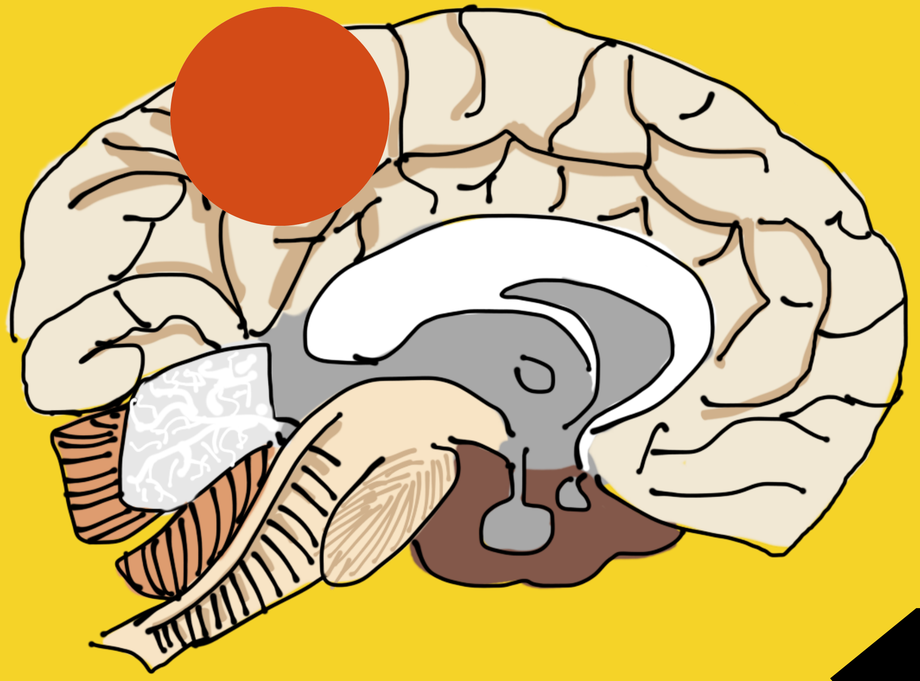
СЧАСТЬЕ



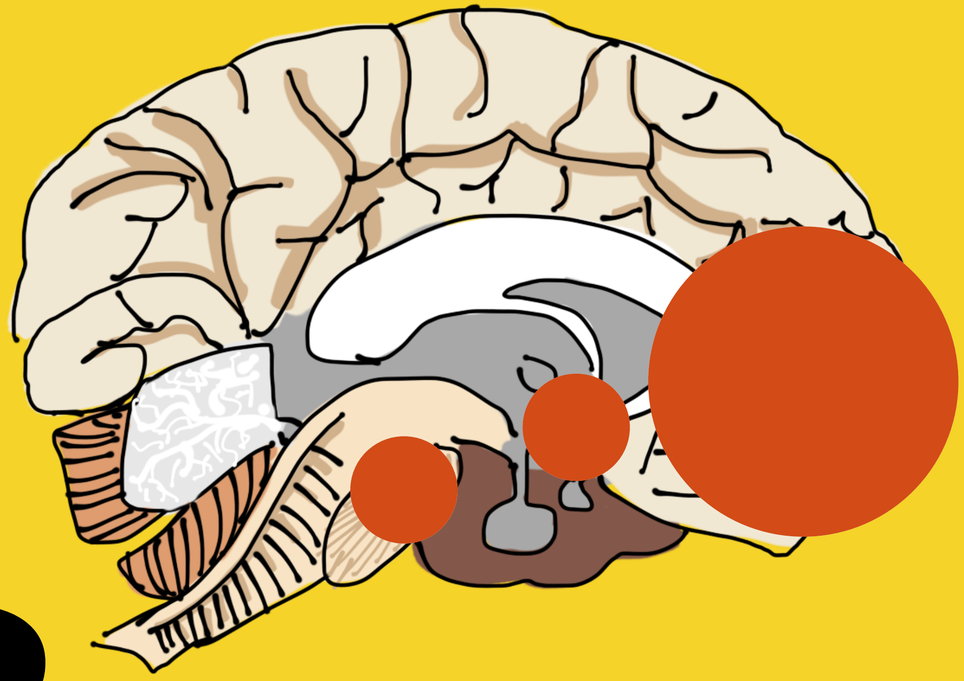
СТРЕСС



СЧАСТЬЕ

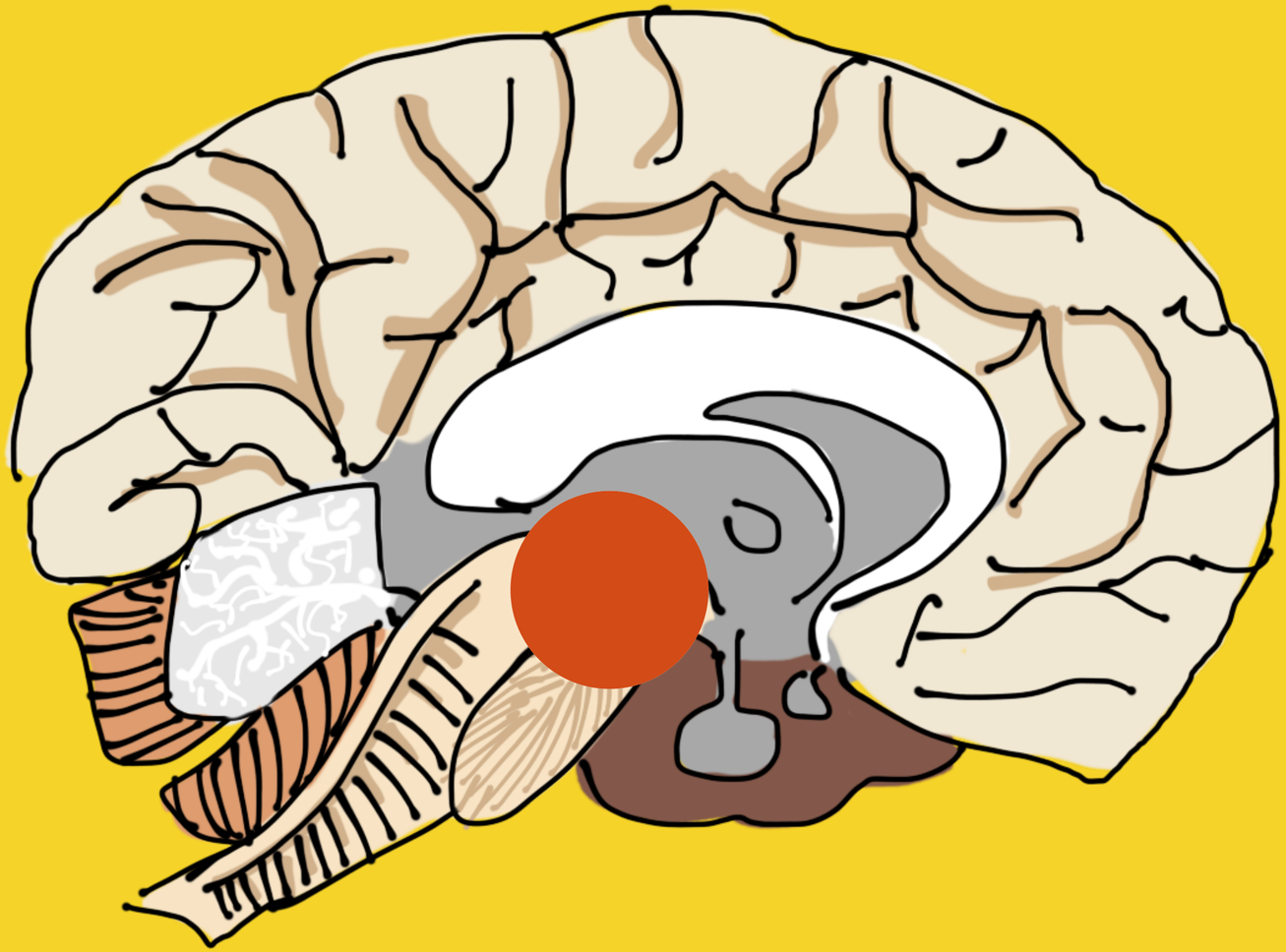


СТРЕСС



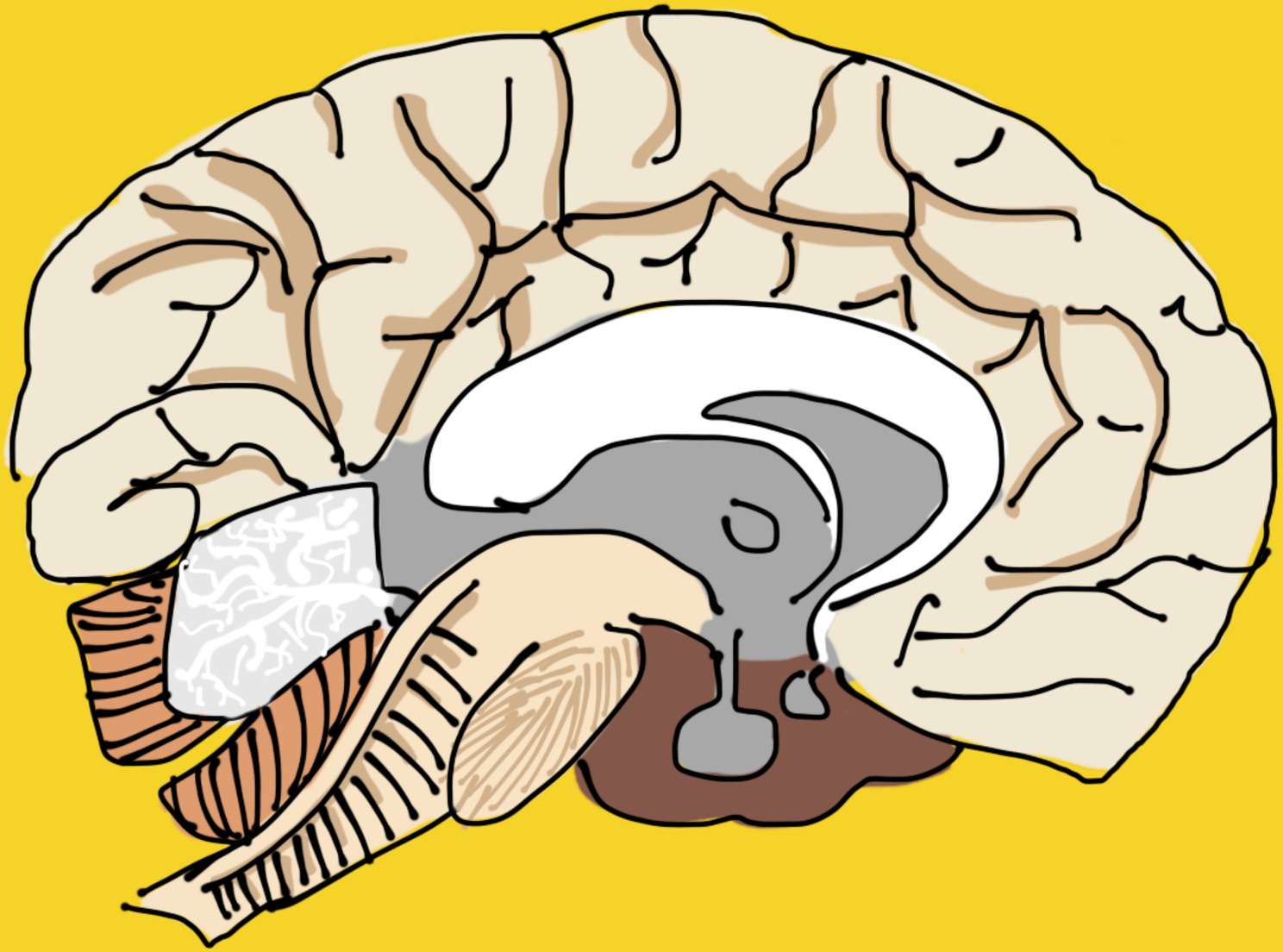
1:3

гиппокамп



гиппокамп

- концентрация внимания
- способность к обучению
- память



Эй, ты!

ПРРР

“**90%** чего угодно —
ПОЛНАЯ ЧУШЬ”

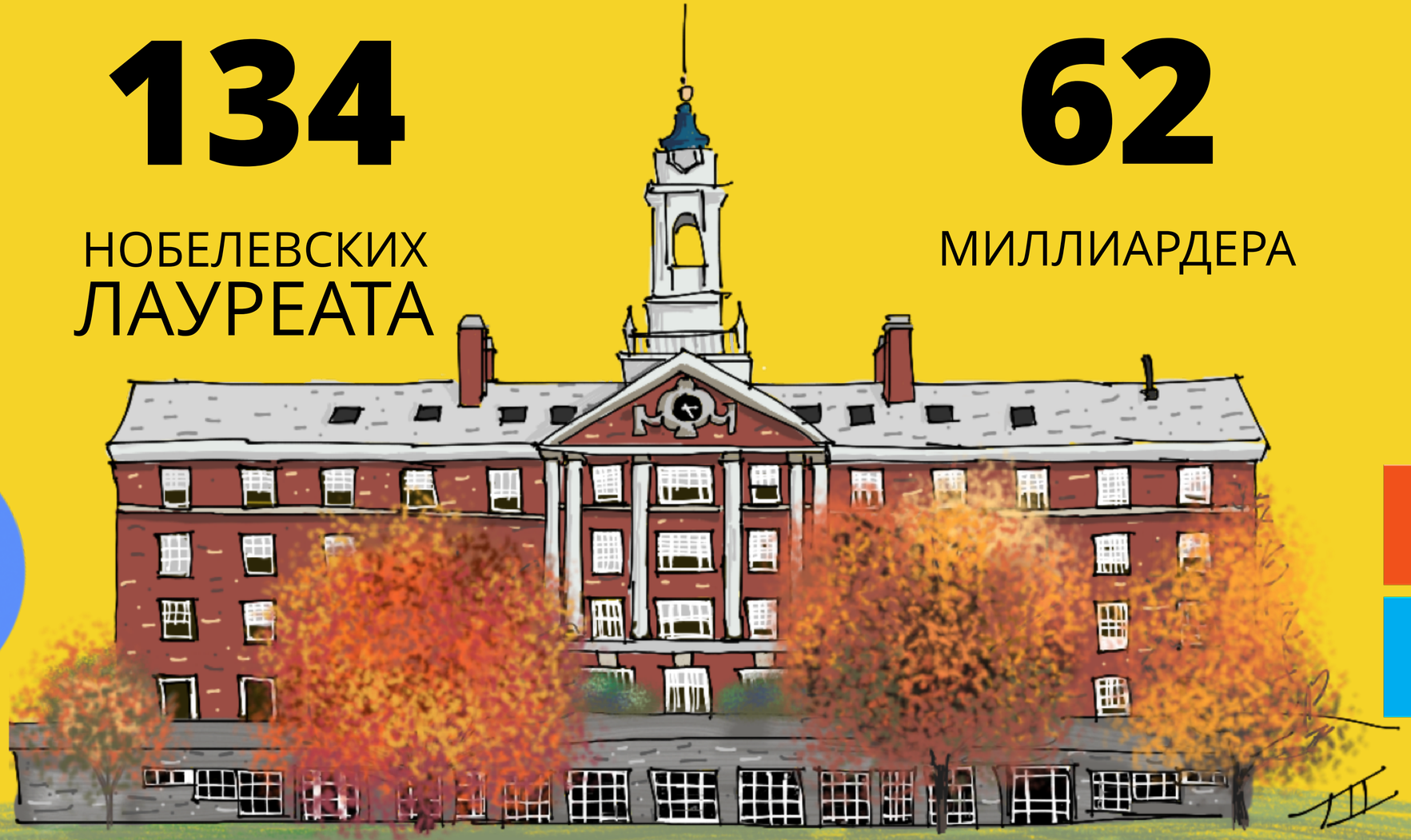
— закон/откровение Старджона —

134

НОБЕЛЕВСКИХ
ЛАУРЕАТА

62

МИЛЛИАРДЕРА



HARVARD
UNIVERSITY
EST. 1636

HARVARD
COLLEGE

Harry R. Lewis
Gordon McKay Professor
of Computer Science
Dean of Harvard College

University Hall, Harvard College
Cambridge, MA 02138
E-mail: lewis@harvard.edu
Phone: (617) 495-1555
FAX: (617) 496-8268

SLOW DOWN

Getting More out of Harvard by Doing Less

Dear Harvard student,

Students arriving at Harvard have gained admission by participating and excelling in a variety of academic and nonacademic activities in their secondary schools. We hope that you will continue to cultivate many of the qualities that distinguished you in your precollege years — your pursuit of excellence, your strength of character, and your ability to balance your academic drive with participation and success in extracurricular activities.

And yet college is different from high school in important ways, and some habits acquired in anticipation of applying to college may not serve you as well while you are here. You may succeed more fully at the things that will be most important to you if you enter Harvard with an open mind about the possibilities available to you, but gradually spend more of your time on fewer things you discover you truly love. You may balance your life better if you participate in some activities purely for fun, rather than to achieve a leadership role that you hope might be a distinctive credential for postgraduate employment. The human relationships you form in unstructured time with your roommates and friends may have a stronger influence on your later life than the content of some of the courses you are taking.

This letter offers some suggestions about how to get the most out of Harvard. Each suggestion requires making choices, which may be hard choices, between doing more things and leaving some possibilities aside. In a larger sense, these suggestions are meant to start you towards a fulfilling life after college, perhaps many years after you leave here. In high school one's academic choices are limited, and most Harvard students have taken the most demanding choice available where there was any choice at all. Many high schools have counseled students that a longer list of activities, with as many leadership roles as possible, would impress college admissions committees more than a shorter list with fewer titles. Yet in later life most of what we do outside our jobs we do because we want to do it, not because we are in any tangible way rewarded for doing it. College is a transition period; we will certainly give you grades and transcripts attesting to some of the things you have done here, but much of what you

КАК?

ПОЧЕМУ?

```
    return path.slice(0, dotIndex);
  },
  isDeep: function (path) {
    return path.indexOf('.') !== -1;
  },
  isAncestor: function (base, path) {
    return base.indexOf(path + '.') === 0;
  },
  isDescendant: function (base, path) {
    return path.indexOf(base + '.') === 0;
  },
  translate: function (base, newBase, path) {
    return newBase + path.slice(base.length);
  },
  matches: function (base, wildcard, path) {
    return base === path || this.isAncestor(base, path) || Boolean(wildcard) && this.isDescendant(base, path);
  }
};Life.Base._addFeature({
  prepAnnotations: function () {
```

Elements Network Performance Console

3 1

✕ ▶	Frustration: repeat count must be less than infinity	life.html:1300
✕ ▶	24/7: BRAIN is not a constructor	life.html:1315
✕ ▶	Perfectionism: too much recursion	life.html:1316
⚠ ▶	Narrow-mindedness: unreachable code after return statement	life.html:1325

>

Perfectionism: too much recursion

https://habrahabr.ru/company/pvs-studio/blog/118204/

Разное → Пора завязывать использовать символы табуляции в коде

Блог компании PVS-Studio

Многие могут счесть спор, о том, что лучше пробелы или табуляции в коде за holy wars. Однако нет, я не хочу устраивать дискуссию на эту тему. Я однозначно утверждаю, что в обязательном порядке следует использовать пробелы. И разговор и «предпочтении того или иного» здесь не уместен. Как не уместно в наше время обсуждать, что удобнее, компьютер или печатная машинка. Поскольку печатные машинки закончили свое существование, ориентироваться в дальнейшем на их использование, по меньшей мере, нерационально. А если ещё учесть, сколь удобнее пользоваться компьютером для набора текста, то вопрос выбора просто отпадает.

С пробелами и табуляцией ситуация не настолько грандиозна и масштабна, но аналогия прослеживается. Далее я поясню, почему рационально перейти на пробелы для форматирования кода.

Эту заметку меня побудило написать то, что табуляции мне надоели. Вроде и мелочь. Но идут годы, а эта инфекция у программистов всё никак не проходит. Хотя сам я при написании кода не использую табуляции, тем не менее, регулярно вспоминаю недобрым словом их поклонников. Не волнуйтесь, от

https://habrahabr.ru/company/pvs-studio/blog/118204/

Комментарии (217)

TheShock 27 апреля 2011 в 10:07 # +61

Вы говорите глупости потому-что не умеете пользоваться табуляцией.
Ненавижу пробелы там, где логически должна быть табуляция (поддерживал несколько проектов с такими Style Guide — чут ьне обблевался).
Больше людей предпочитают табы вместо пробелов.
А то, что вы выдаете за истину — ошибочное утверждение от незнания.

widowmaker 27 апреля 2011 в 10:12 # ↰ ↑ +14

Объясняйте свою позицию, а не просто «блюю от пробелов», и «вы просто не умеете их(табы) готовить»

TheShock 27 апреля 2011 в 10:18 # ↰ ↑ +1

Ок, сейчас напишу развернутый ответ.

TheShock 27 апреля 2011 в 10:44 # ↰ ↑ +28

Как и просили)
Пора завязывать использовать пробелы вместо табуляции в коде

widowmaker 27 апреля 2011 в 11:38 # ↰ ↑ –6

Спасибо за достойный ответ :)

- снижение КПД
- паралич перфекционизма
- решение необнаруженных проблем

АДЕКВАТНЫЙ
перфекционизм

“В аду для перфекционистов
Ни серы нет, и нет огня.
А лишь слегка асимметрично
стоят щербатые котлы.”

— Неизвестный (мне) автор —

```
    return path.slice(0, dotIndex);
  },
  isDeep: function (path) {
    return path.indexOf('.') !== -1;
  },
  isAncestor: function (base, path) {
    return base.indexOf(path + '.') === 0;
  },
  isDescendant: function (base, path) {
    return path.indexOf(base + '.') === 0;
  },
  translate: function (base, newBase, path) {
    return newBase + path.slice(base.length);
  },
  matches: function (base, wildcard, path) {
    return base === path || this.isAncestor(base, path) || Boolean(wildcard) && this.isDescendant(base, path);
  }
};Life.Base._addFeature({
  prepAnnotations: function () {
```

Elements Network Performance Console

3 1

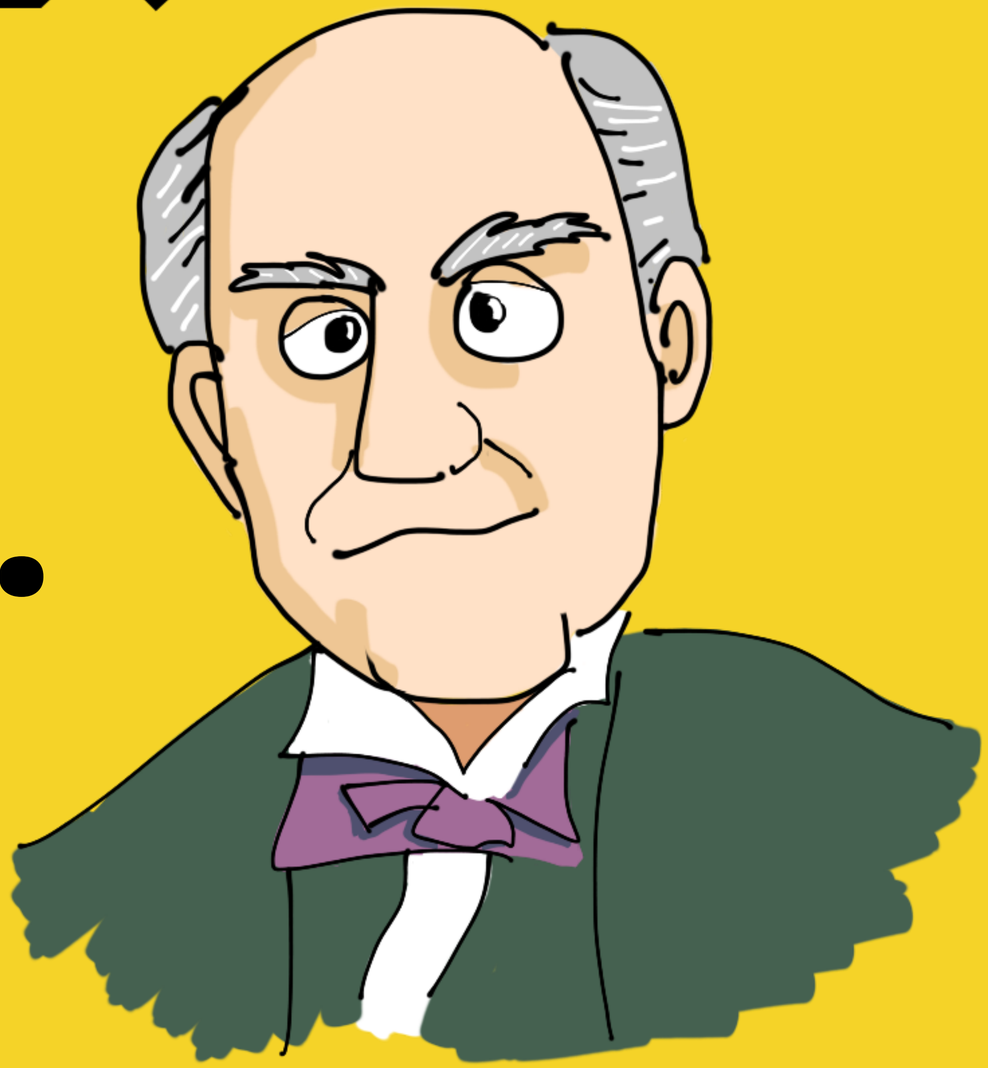
✕ ▶	Frustration: repeat count must be less than infinity	life.html:1300
✕ ▶	24/7: BRAIN is not a constructor	life.html:1315
✕ ▶	Perfectionism: too much recursion	life.html:1316
⚠ ▶	Narrow-mindedness: unreachable code after return statement	life.html:1325

>

⚠ ▶ Narrow-mindedness: unreachable code after return statement

AC/DC

vs.



I vs. T

Сам доклад поставлен на отлично.
У меня один вопрос: …
Почему** на технической организации **мы говорим
***о психологии пользователя**?*

— Неизвестный (мне) автор —

БЮДЖЕТ
ПРОИЗВОДИТЕЛЬНОСТИ

ФИЧА

БЮДЖЕТ

ПРОИЗВОДИТЕЛЬНОСТИ

БЮДЖЕТ

ПРОИЗВОДИТЕЛЬНОСТИ

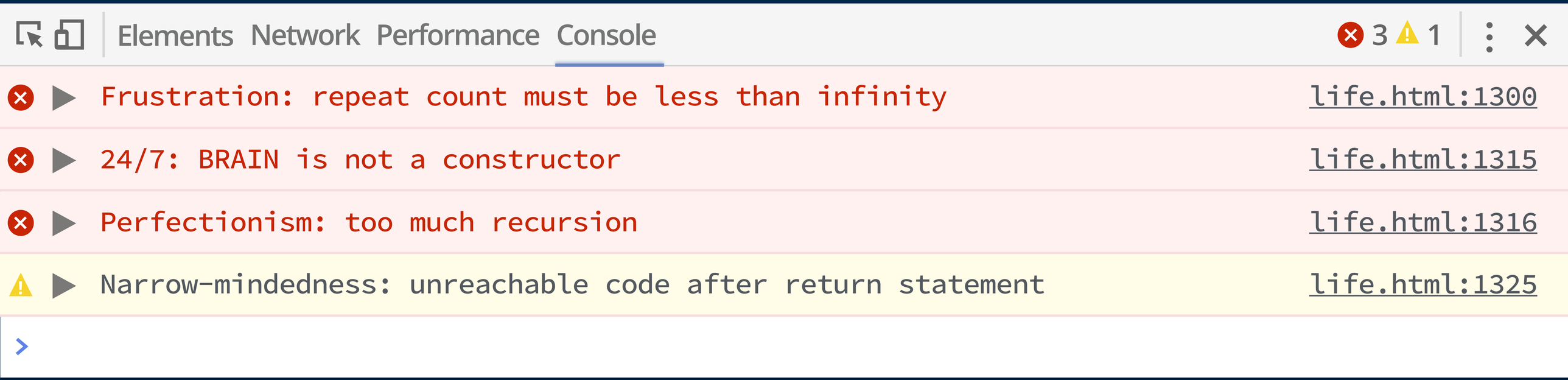
ИЧА

20%

БЮДЖЕТ

ПРОИЗВОДИТЕЛЬНОСТИ

ИЧА


Elements Network Performance Console
3 1
Frustration: repeat count must be less than infinity
life.html:1300
24/7: BRAIN is not a constructor
life.html:1315
Perfectionism: too much recursion
life.html:1316
Narrow-mindedness: unreachable code after return statement
life.html:1325
>

СПАСИБО

ДА, **ИЛЛЮСТРАЦИИ** МОИ:)

DENYS MISHUNOV • DIGITAL GARDEN AS **TWITTER:** @MISHUNOV

Elements Network Performance Console 3 1

```
Frustration: repeat count must be less than infinity   life.html:1300
24/7: BRAIN is not a constructor                       life.html:1315
Perfectionism: too much recursion                      life.html:1316
Narrow-mindedness: unreachable code after return statement   life.html:1325
>
```