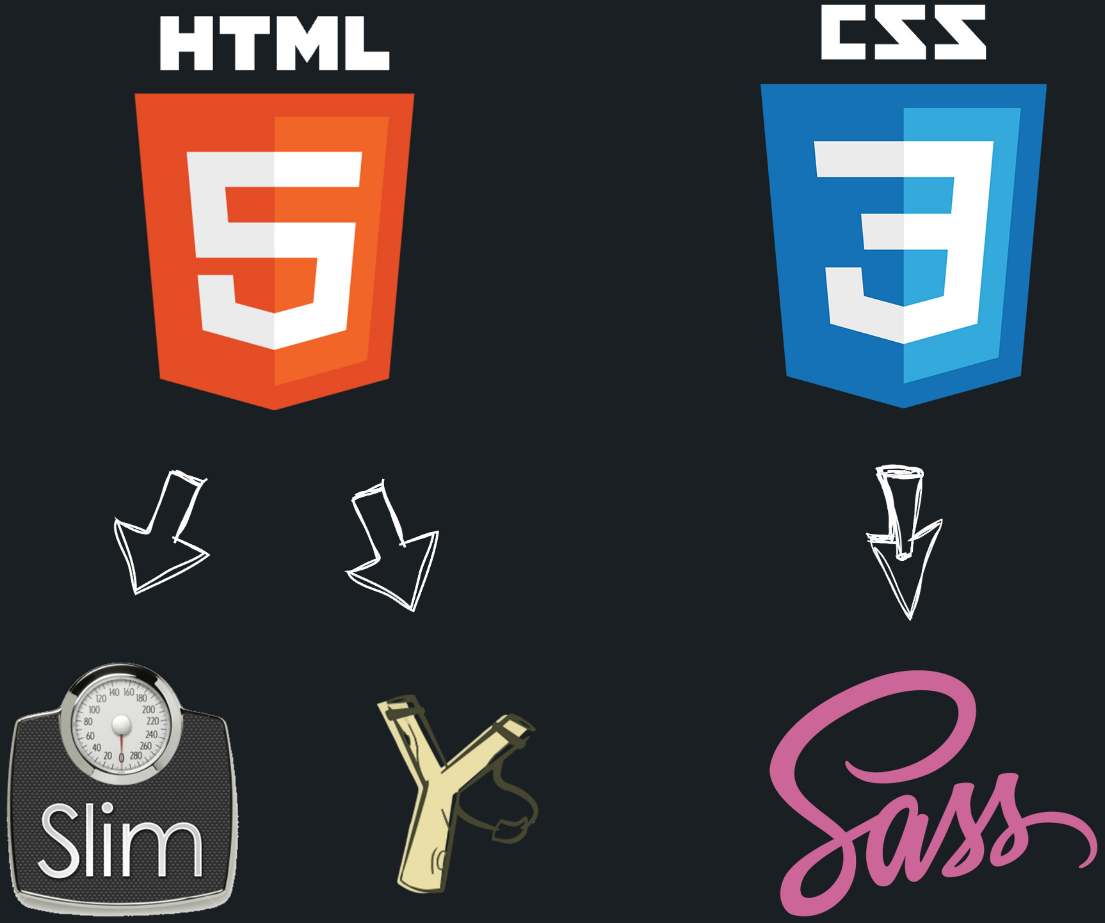
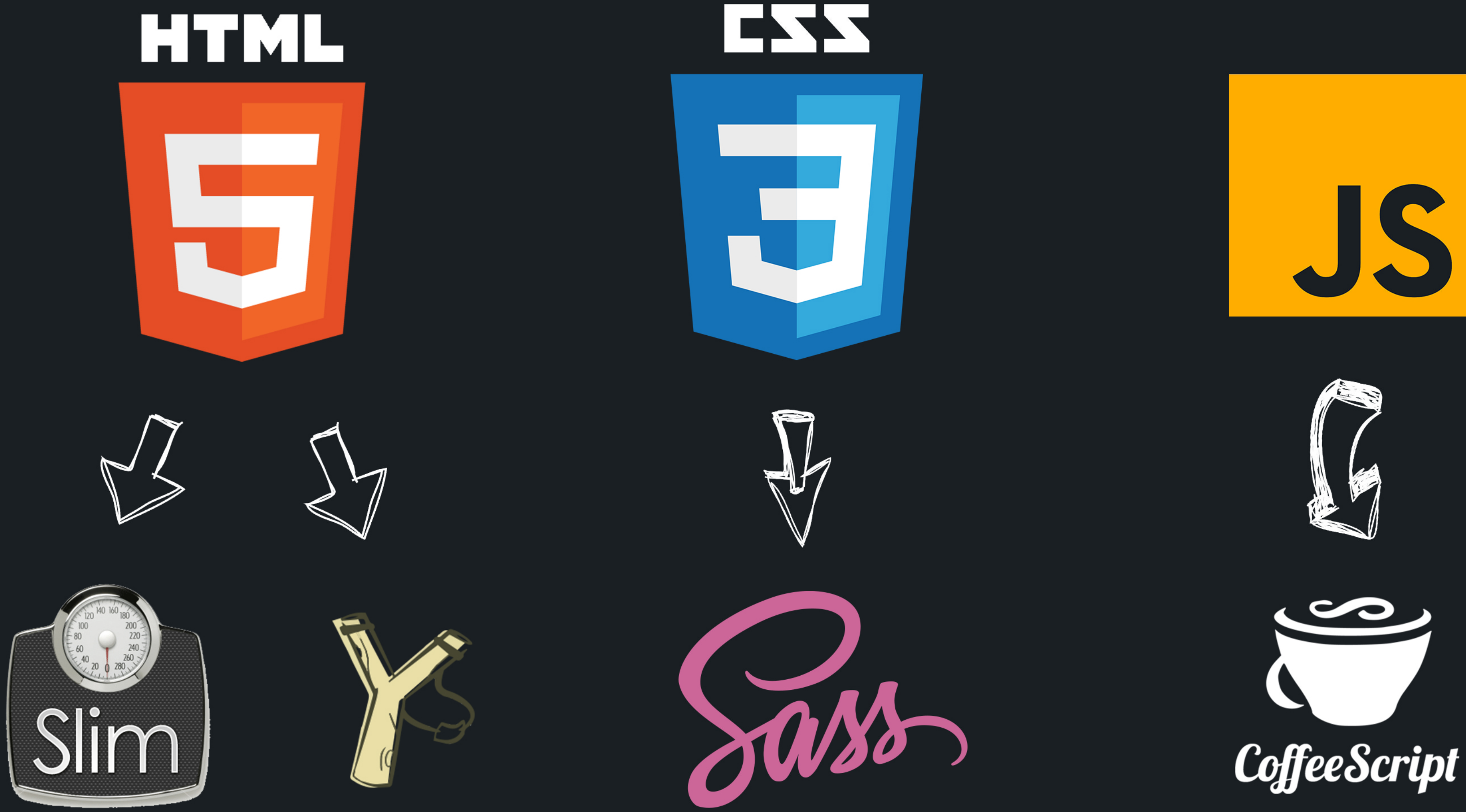
FRONTEND DEV TOOLS

Intro

Intro



Intro



HTML VS HAML & SLIM

HTML

```html
<section id="users">
  <div class="user" data-user-id="45">
    <strong class="user__nickname">
      czajkovsky
    </strong>
    Some info about this user.
  </div>
</section>
```

HTML VS HAML & SLIM

HTML

```html
<section id="users">
  <div class="user" data-user-id="45">
    <strong class="user__nickname">
      czajkovsky
    </strong>
    Some info about this user.
  </div>
</section>
```

HAML

```haml
%section#users
  .user{ data: { user_id: 45 } }
    %strong.user__nickname czajkovsky
    Some info about this user.
```

HTML VS HAML & SLIM

HTML

```
<section id="users">
  <div class="user" data-user-id="45">
    <strong class="user__nickname">
      czajkovsky
    </strong>
    Some info about this user.
  </div>
</section>
```

HAML

```
%section#users
  .user{ data: { user_id: 45 } }
    %strong.user__nickname czajkovsky
    Some info about this user.
```

SLIM

```
section#users
  .user data-user-id=45
    strong.user__nickname czajkovsky
    | Some info about this user.
```

HTML VS HAML & SLIM

HTML

```html
<section id="users">
  <div class="user" data-user-id="45">
    <strong class="user__nickname">
      czajkovsky
    </strong>
    Some info about this user.
  </div>
</section>
```

HAML

```haml
%section#users
  .user{ data: { user_id: 45 } }
    %strong.user__nickname czajkovsky
    Some info about this user.
```

SLIM

```slim
section#users
  .user data-user-id=45
    strong.user__nickname czajkovsky
    | Some info about this user.
```

PROS

enforces on developer correct indent

you don’t have to remember about closing tags

code is much shorter

HTML VS HAML & SLIM

HTML

```html
<section id="users">
  <div class="user" data-user-id="45">
    <strong class="user__nickname">
      czajkovsky
    </strong>
    Some info about this user.
  </div>
</section>
```

HAML

```haml
%section#users
  .user{ data: { user_id: 45 } }
    %strong.user__nickname czajkovsky
    Some info about this user.
```

SLIM

```slim
section#users
  .user data-user-id=45
    strong.user__nickname czajkovsky
    | Some info about this user.
```

PROS

enforces on developer correct indent

you don't have to remember about closing tags

code is much shorter

CONS?

new developer must be familiar with it
(learning takes 20 minutes)

HTML VS HAML & SLIM

HTML

```
<section id="users">
  <div class="user" data-user-id="45">
    <strong class="user__nickname">
      czajkovsky
    </strong>
    Some info about this user.
  </div>
</section>
```

HAML

```
%section#users
  .user{ data: { user_id: 45 } }
    %strong.user__nickname czajkovsky
    Some info about this user.
```

SLIM

```
section#users
  .user data-user-id=45
    strong.user__nickname czajkovsky
    | Some info about this user.
```

PROS

enforces on developer correct indent

you don't have to remember about closing tags

code is much shorter

CONS?

new developer must be familiar with it (learning takes 20 minutes)

HAML VS SLIM

Slim is faster

HAML is more popular

Slim allows you to define custom shortcuts

CSS vs Sass (SCSS)

CSS

```css
.users {
  background: #fff;
}
.users h3 {
  color: #000;
}
```

CSS vs Sass (SCSS)

CSS

```css
.users {
  background: #fff;
}
.users h3 {
  color: #000;
}
```

SCSS

```scss
.users {
  background: #fff;
  h3 {
    color: #000;
  }
}
```

CSS vs Sass (SCSS)

CSS

```css
.users {
  background: #fff;
}
.users h3 {
  color: #000;
}
```

SCSS

```scss
.users {
  background: #fff;
  h3 {
    color: #000;
  }
}
```

Sass

```sass
.users
  background: #fff;
  h3
    color: #000;
```

CSS vs Sass (SCSS)

CSS

```css
.users {
  background: #fff;
}
.users h3 {
  color: #000;
}
```

SCSS

```scss
.users {
  background: #fff;
  h3 {
    color: #000;
  }
}
```

Sass

```sass
.users
  background: #fff;
  h3
    color: #000;
```

Pros

code is more DRY, much shorter and easier to maintain

CSS vs **Sass** (**SCSS**)

CSS

```
.users {
  background: #fff;
}
.users h3 {
  color: #000;
}
```

SCSS

```
.users {
  background: #fff;
  h3 {
    color: #000;
  }
}
```

Sass

```
.users
  background: #fff;
  h3
    color: #000;
```

Pros

code is more DRY, much shorter and easier to maintain

Cons

no cons :)

CSS vs **Sass** (**SCSS**)

CSS

```css
.users {
  background: #fff;
}
.users h3 {
  color: #000;
}
```

SCSS

```scss
.users {
  background: #fff;
  h3 {
    color: #000;
  }
}
```

Sass

```sass
.users
  background: #fff;
  h3
    color: #000;
```

Pros

code is more DRY, much shorter and easier to maintain

Cons

no cons :)

Sass vs SCSS

Sass has more concise syntax
(doesn’t require curly brackets, semicolons)

SCSS is compatible with CSS

Sass variables & reference symbol

Variables

SCSS

```scss
$width: 1rem;
.foo {
  width: $width;
  height: $width;
}
```

CSS

```css
.foo {
  width: 1rem;
  height: 1rem;
}
```

Sass variables & reference symbol

Variables

SCSS

```scss
$width: 1rem;
.foo {
  width: $width;
  height: $width;
}
```

CSS

```css
.foo {
  width: 1rem;
  height: 1rem;
}
```

Reference symbol

SCSS

```scss
.foo {
  color: #f00;
  &.bar {
    height: #000;
  }
}
```

CSS

```css
.foo {
  color: #f00;
}
.foo.bar {
  color: #000;
}
```

Sass variables & reference symbol

Variables

SCSS

```scss
$width: 1rem;
.foo {
  width: $width;
  height: $width;
}
```

CSS

```css
.foo {
  width: 1rem;
  height: 1rem;
}
```

Reference symbol

SCSS

```scss
.foo {
  color: #f00;
  &.bar {
    height: #000;
  }
}
```

CSS

```css
.foo {
  color: #f00;
}
.foo.bar {
  color: #000;
}
```

SCSS

```scss
.foo {
  color: #f00;
  .ie7 & {
    height: #000;
  }
}
```

CSS

```css
.foo {
  color: #f00;
}
.ie7 .foo {
  color: #000;
}
```

Sass BEM syntax

Reference symbol - BEM syntax (Sass 3.3)

SCSS

```scss
.user {
  &__nickname {
    color: #f00;
  }
  &--unverified {
    background: #00f;
  }
}
```

CSS

```css
.user__nickname {
  color: #f00;
}
.user--unverified {
  background: #00f;
}
```

Sass BEM syntax

Reference symbol - BEM syntax (Sass 3.3)

SCSS

```scss
.user {
  &__nickname {
    color: #f00;
  }
  &--unverified {
    background: #00f;
  }
}
```

CSS

```css
.user__nickname {
  color: #f00;
}
.user--unverified {
  background: #00f;
}
```

Pros

CSS reflects HTML structure

easier to read

Cons

HTML classes names can be long

Sass mixins & functions

Mixins

SCSS

```scss
@mixin square($size) {
  width: $size;
  height: $size;
}

.post {
  @include square(1rem);
}
```

CSS

```css
.post {
  width: 1rem;
  height: 1rem;
}
```

Sass mixins & functions

Mixins

SCSS

```scss
@mixin square($size) {
  width: $size;
  height: $size;
}

.post {
  @include square(1rem);
}
```

CSS

```css
.post {
  width: 1rem;
  height: 1rem;
}
```

Functions

SCSS

```scss
@function double($value) {
  @return $value * 2;
}

.post {
  width: double(14px);
}
```

CSS

```css
.post {
  width: 28px;
}
```

Sass @extend, placeholders

@extend the bad way

SCSS

```scss
.button {
  border-color: #f00;
}

.button--big {
  @extend .button;
  background: #00f;
}
```

CSS

```css
.button, .button--big {
  border-color: #f00;
}

.button--big {
  background: #00f;
}
```

Sass @extend, placeholders

@extend the bad way

SCSS

```scss
.button {
  border-color: #f00;
}

.button--big {
  @extend .button;
  background: #00f;
}
```

CSS

```css
.button, .button--big {
  border-color: #f00;
}

.button--big {
  background: #00f;
}
```

Placeholders

SCSS

```scss
%button {
  border-color: #f00;
}

.button--big {
  @extend %button;
  background: #00f;
}
```

CSS

```css
.button--big {
  border-color: #f00;
}

.button--big {
  background: #00f;
}
```

Sass ifs, loops & maps

If

```
.post {
  @if ($dark == true) {
    color: #000;
  } @else {
    color: #fff;
  }
}
.post {
  color: if($dark == true, #000, #fff);
}
```

Sass ifs, loops & maps

If

```scss
.post {
  @if ($dark == true) {
    color: #000;
  } @else {
    color: #fff;
  }
}
.post {
  color: if($dark == true, #000, #fff);
}
```

For

```scss
@for $i from 3 through 1 {
  h#{4 - $i} {
    font-size: 2rem * $i;
  }
}
```

Sass ifs, loops & maps

If

```
.post {
  @if ($dark == true) {
    color: #000;
  } @else {
    color: #fff;
  }
}
.post {
  color: if($dark == true, #000, #fff);
}
```

Each

```
$alerts: (error, red, 1px),
         (success, green, 2px),
         (info, blue, 2px);
@each $type, $color, $border in $alerts {
  .alert--#{$type} {
    border: $border solid $color;
  }
}
```

For

```
@for $i from 3 through 1 {
  h#{4 - $i} {
    font-size: 2rem * $i;
  }
}
```

Sass ifs, loops & maps

If

```scss
.post {
  @if ($dark == true) {
    color: #000;
  } @else {
    color: #fff;
  }
}
.post {
  color: if($dark == true, #000, #fff);
}
```

For

```scss
@for $i from 3 through 1 {
  h#{4 - $i} {
    font-size: 2rem * $i;
  }
}
```

Each

```scss
$alerts: (error, red, 1px),
         (success, green, 2px),
         (info, blue, 2px);
@each $type, $color, $border in $alerts {
  .alert--#{$type} {
    border: $border solid $color;
  }
}
```

Maps

```scss
$indexes: (
  menu:  999,
  badge: 1050
);

.foo {
  z-index: map-get($indexes, menu);
}
```

Sass @content directive

@content directive

SCSS

```scss
@mixin media($device, $only: false) {
  ...
  @media screen and (min-width: $min-width) {
    @content;
  }
  ...
}
// full mixin https://netguru.co/blog/categories/css

.foo {
  @include media(tablet, true) {
    background: #f00;
  }
}
```

Sass @content directive

@content directive

SCSS

```scss
@mixin media($device, $only: false) {
  ...
  @media screen and (min-width: $min-width) {
    @content;
  }
  ...
}
// full mixin https://netguru.co/blog/categories/css

.foo {
  @include media(tablet, true) {
    background: #f00;
  }
}
```

CSS

```css
@media screen and (min-width: 48rem) and (max-width: 62rem) {
  .foo {
    background: #f00;
  }
}
```

CoffeeScript

```
class Lecture

  constructor: (@title, @author) ->
    @slides = []

  display_date: ->
    if @date? then @date else "Sorry, no info about date for #{@title}"

  add_slide: (name, duration) ->
    slide =
      name: name
      duration: duration
    @slides.push slide

  long_slides: ->
    s.name for s in @slides when s.duration > 3
```

CoffeeScript

```
class Lecture

  constructor: (@title, @author) ->
    @slides = []

  display_date: ->
    if @date? then @date else "Sorry, no info about date for #{@title}"

  add_slide: (name, duration) ->
    slide =
      name: name
      duration: duration
    @slides.push slide

  long_slides: ->
    s.name for s in @slides when s.duration > 3
```

Pros

code is shorter by 1/3 without any incluence on execution time (is compiled to JS)

syntax (default in RoR from version 3.1)

CoffeeScript

```
class Lecture

  constructor: (@title, @author) ->
    @slides = []

  display_date: ->
    if @date? then @date else "Sorry, no info about date for #{@title}"

  add_slide: (name, duration) ->
    slide =
      name: name
      duration: duration
    @slides.push slide

  long_slides: ->
    s.name for s in @slides when s.duration > 3
```

Pros

code is shorter by 1/3 without any incluence on execution time
(is compiled to JS)

syntax (default in RoR from version 3.1)

Cons

syntax

CoffeeScript

@ Alias

```
@id # this.id
class Lecture
  @findByTitle: (title) ->
    ...
```

CoffeeScript

@ Alias

```
@id # this.id
class Lecture
  @findByTitle: (title) ->
    ...
```

Existential operator

```
'foo exists' if foo?
# typeof foo !== "undefined" &&
# foo !== null

foo?.prop?.subprop?
foo.prop ?= 'new val'
```

CoffeeScript

@ Alias

```
@id # this.id
class Lecture
  @findByTitle: (title) ->
    ...
```

Existential operator

```
'foo exists' if foo?
# typeof foo !== "undefined" &&
# foo !== null

foo?.prop?.subprop?
foo.prop ?= 'new val'
```

Objects, arrays

```
# foo = { bar: 1, baz: 2 }
foo =
  bar: 1
  baz: 2

# range = [2, 3, 4, 5]
range = [2..5]
```

CoffeeScript

@ Alias

```
@id # this.id
class Lecture
  @findByTitle: (title) ->
    ...
```

Existential operator

```
'foo exists' if foo?
# typeof foo !== "undefined" &&
# foo !== null

foo?.prop?.subprop?
foo.prop ?= 'new val'
```

Objects, arrays

```
# foo = { bar: 1, baz: 2 }
foo =
  bar: 1
  baz: 2

# range = [2, 3, 4, 5]
range = [2..5]
```

Loops & comprehensions

```
for name, i in ['HAML', 'CSS', 'JS']
  alert "#{i}: #{name}"

# a = [{ name: 'HAML', duration: 5 }, ...]
s.name for s in a when s.duration > 3
```

CoffeeScript

Functions

```coffeescript
foo = (bar) ->
  # your function body

square = (x) -> x * x

power = (a = 1, b = 2) ->
  Math.pow(a, b)
```

CoffeeScript

Functions

```
foo = (bar) ->
  # your function body

square = (x) -> x * x

power = (a = 1, b = 2) ->
  Math.pow(a, b)
```

Auto return

```
foo: ->
  # your function body
  "this will be returned"
```

CoffeeScript

Functions

```
foo = (bar) ->
  # your function body

square = (x) -> x * x

power = (a = 1, b = 2) ->
  Math.pow(a, b)
```

Flow control

```
'boring...' if sleeping_folks > 2
unless foo is 1
if bar isnt 2 and baz is 7
'mid-length' if 3 < l.duration < 5
```

Auto return

```
foo: ->
  # your function body
  "this will be returned"
```

CoffeeScript

Functions

```coffeescript
foo = (bar) ->
  # your function body

square = (x) -> x * x

power = (a = 1, b = 2) ->
  Math.pow(a, b)
```

Auto return

```coffeescript
foo: ->
  # your function body
  "this will be returned"
```

Flow control

```coffeescript
'boring...' if sleeping_folks > 2
unless foo is 1
if bar isnt 2 and baz is 7
'mid-length' if 3 < l.duration < 5
```

Interpolation

```coffeescript
"#{title} by #{author}" # OK
'#{title} by #{author}' # won't work
```

CoffeeScript

Fat arrow

```
class Lecture

  prepare: (@subject) ->

  notify: ->
    alert @subject()

class Speaker

  constructor: (@name) ->
    @lecture = new Lecture
    @lecture.prepare () => "New lecture by #{@name}"

p = new Speaker('czajkovsky')
p.lecture.notify()
```

CoffeeScript

Fat arrow

```
class Lecture

  prepare: (@subject) ->

  notify: ->
    alert @subject()

class Speaker

  constructor: (@name) ->
    @lecture = new Lecture
    @lecture.prepare () => "New lecture by #{@name}"

p = new Speaker('czajkovsky')
p.lecture.notify()
```

Multiple assignment

```
[month, day, year] = 'June 2 14'.split ' '
```

CoffeeScript

```
class Lecture

  constructor: (@title, @author) ->
    @slides = []

  display_date: ->
    if @date? then @date else "Sorry, no info about date for #{@title}"

  long_slides: ->
    s.name for s in @slides when s.duration > 3

  add_slide: (name, duration) ->
    slide =
      name: name
      duration: duration
    @slides.push slide
```

CoffeeScript

```
class Lecture

  constructor: (@title, @author) ->
    @slides = []

  display_date: ->
    if @date? then @date else "Sorry, no info about date for #{@title}"

  long_slides: ->
    s.name for s in @slides when s.duration > 3

  add_slide: (name, duration) ->
    slide =
      name: name
      duration: duration
    @slides.push slide
```

```
l = new Lecture('Front-end tools', 'czajkovsky')
l.display_date()
# Sorry, no info about date for Front-end tools
```

CoffeeScript

```coffeescript
class Lecture

  constructor: (@title, @author) ->
    @slides = []

  display_date: ->
    if @date? then @date else "Sorry, no info about date for #{@title}"

  long_slides: ->
    s.name for s in @slides when s.duration > 3

  add_slide: (name, duration) ->
    slide =
      name: name
      duration: duration
    @slides.push slide
```

```coffeescript
l = new Lecture('Front-end tools', 'czajkovsky')
l.display_date()
# Sorry, no info about date for Front-end tools

l.date = '02/06/2014 18:30'
l.display_date()
# 02/06/2014 18:30
```

CoffeeScript

```coffeescript
class Lecture

  constructor: (@title, @author) ->
    @slides = []

  display_date: ->
    if @date? then @date else "Sorry, no info about date for #{@title}"

  long_slides: ->
    s.name for s in @slides when s.duration > 3

  add_slide: (name, duration) ->
    slide =
      name: name
      duration: duration
    @slides.push slide
```

```coffeescript
l = new Lecture('Front-end tools', 'czajkovsky')
l.display_date()
# Sorry, no info about date for Front-end tools

l.date = '02/06/2014 18:30'
l.display_date()
# 02/06/2014 18:30

l.add_slide('HAML', 2)
l.add_slide('Sass', 5)
l.add_slide('CoffeeScript', 4)
l.long_slides()
# ["Sass", "CoffeeScript"]
```