

CSS **AND BEERS** IS AWESOME

Una proposta una mica sensata per mantenir
una arquitectura escalable del CSS en els
projectes de Calidae.

Les preguntes ens agraden!

CSS escalable?

CSS fàcil d'entendre i de mantenir?

CSS fàcil de revisar?

CSS comú en diversos projectes?

(Això existeix...?)

Què tenim?



ESS

Global
HD



Què tenim (realment)?

Cascada (La “C” de CSS)

Especificitat

Un sol document (DOM)

Com ho solucionem?

“Convention over configuration”

O sigui: reduir el nombre de decisions que un desenvolupador ha de prendre.

Arribar a acords i mantenir-los.

(Sass)

Arquitectura ITCSS

Namespaces

Nomenclatura BEM

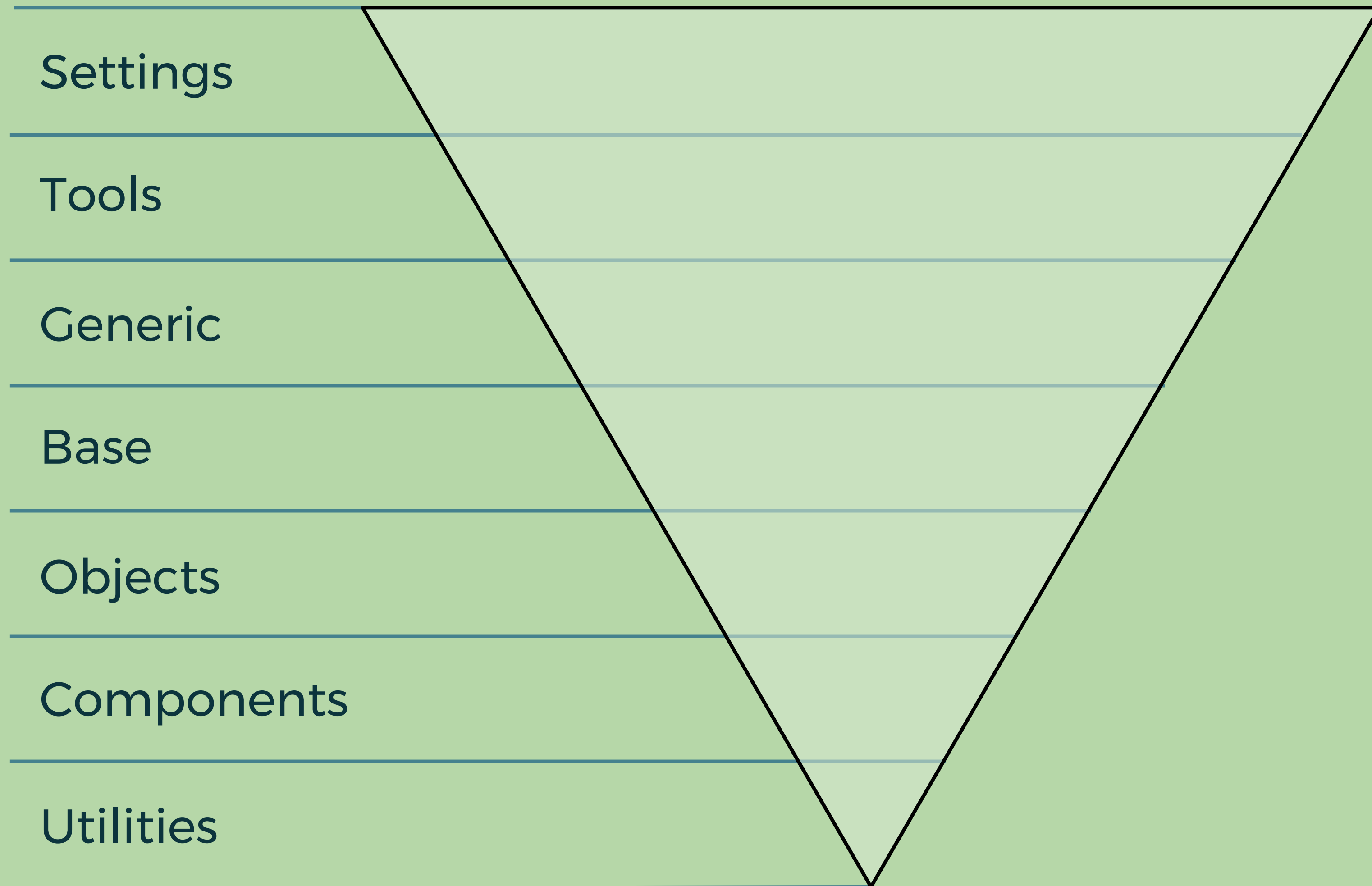
Bones pratiques

Introducing...

ITCSS

Inverted Triangle CSS

Arquitectura ITCSS



Arquitectura ITCSS

◀ SCSS

▶ 1-Settings

▶ 2-Tools

▶ 3-Generic

▶ 4-Base

▶ 5-Objects

▶ 6-Components

▶ 7-Utilities

♀ main.scss

Arquitectura ITCSS

“[ITCSS] is a fully managed CSS architecture. **It’s not a framework or library**; there’s nothing to download or install.

It’s a **collection of principles** and metrics by which developers should group and order their CSS in order to keep it **scalable**, terse, logical, and **manageable**.”

- *Harry Roberts*

1 - Settings

Configuracions globals:
colors, mides, variables...

2 - Tools

Funcions i mixins globals

▲ 1-Settings

🔗 `_main.scss`

🔗 `_settings.colors.scss`

🔗 `_settings.global.scss`

🔗 `_settings.sizes.scss`

▲ 2-Tools

🔗 `_main.scss`

🔗 `_tools.clearfix.scss`

🔗 `_tools.math.scss`

3 - Generic

Estils de molt baix nivell:
Normalize.css, box-sizing...

4 - Base

Estils base per elements
HTML “plans” (<h1>, ...)

3-Generic

- 🔗 `_generic.box-sizing.scss`
- 🔗 `_generic.normalize.scss`
- 🔗 `_generic.reset.scss`
- 🔗 `_main.scss`

4-Base

- 🔗 `_base.heading.scss`
- 🔗 `_base.images.scss`
- 🔗 `_base.page.scss`
- 🔗 `_main.scss`

5 - Objects

Patrons de disseny. Dóna estructura i no estil “cosmètic”.

Simple i reusables fins i tot entre projectes.

Modificar els estils d'un Object és una acció perillosa.

▲ 5-Objects

🔗 `_main.scss`

🔗 `_objects.block.scss`

🔗 `_objects.cover.scss`

🔗 `_objects.media.scss`

🔗 `_objects.pack.scss`

Object: Media

Imatge a l'esquerra, amb contingut a la dreta



[\(Link\)](#)



Calidae @calidae · 24 març

Celebrant el Dinar del Sopar de Nadal Mensual del mes de març a Calidae!



Día 4-5: Mykonos.

Dispones de 2 días completos más (además del de llegada y del de regreso) para disfrutar, de día y de noche, de la isla de Mykonos. Disfruta de sus numerosas playas pero también de los pueblos pintorescos del interior, de la ciudad de Chora con la "pequeña Venecia" y las terrazas junto al mar, visita alguna de las más de 3000 iglesias ortodoxas de la isla, o toma una barca hasta la vecina isla de Delos. Comprobarás en cada momento del día que la luz de Mykonos es mágica. Y por la noche la magia de la isla continúa en uno de los destinos más bulliciosos del verano en el Egeo. En total pasarás 3 noches en Mykonos.

Object: Block

Imatge i continguts apilats (stacked)



A mida

Som experts assessors que inspirem, recomanem i planifiquem els viatges de manera totalment personalitzada.



Path-cloud.com

LOGIN



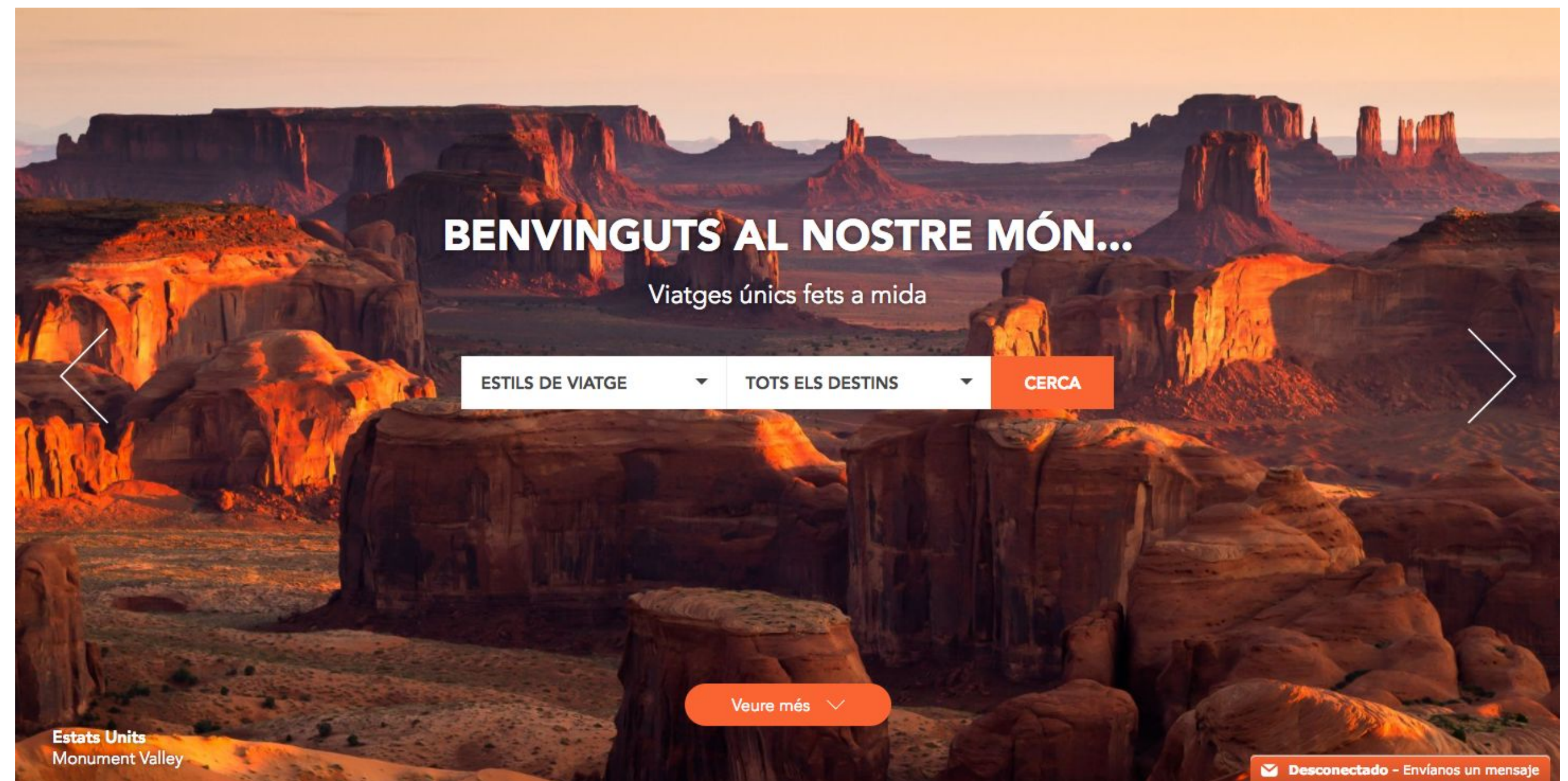
Cirnol comprimidos

Bienestar de la circulación de tus piernas.

[Ver producto](#)

Object: Cover

Imatge com a cover del contingut

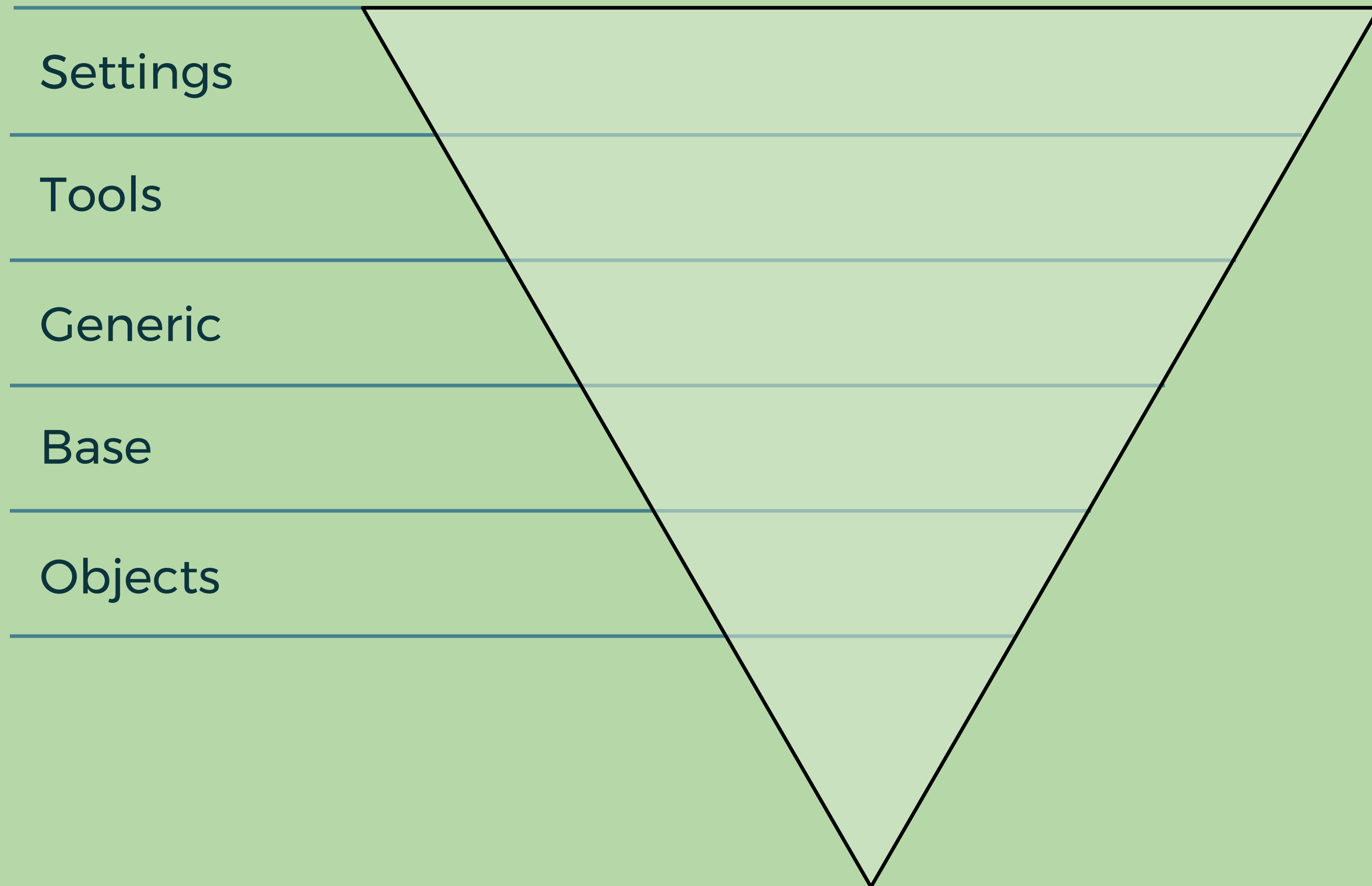


Object: Pack

Elements horitzontals en tota l'amplada disponible



on érem...?



Settings

Tools

Generic

Base

Objects

6 - Components

Implementacions concretes de la nostra UI. Reusables i flexibles.

És on passarem la major part del temps escrivint CSS.

Es poden modificar sense conseqüències inesperades.

▲ 6-Components

- 🔗 `_components.button.scss`
- 🔗 `_components.main-slider.scss`
- 🔗 `_components.modal.scss`
- 🔗 `_components.site-search.scss`
- 🔗 `_main.scss`

Objects per reaprofitament

Heredables pels components.

Un Object està obert a extensions i tancat a modificacions.

Estils per la distribució (layout).

Separen l'estructura de la vista, i els contenidors del contingut.

Components per concretesa

Reusables i flexibles.

Semàntica associada a l'apariència, no al contingut.

Estils cosmètics.

Objects per reaprofitament

display, position, float, margin, padding,
alignment, width, clearfix...

Components per concretesa

colors, borders, font-size, font-family..
(i distribució d'elements fills!)

7 - Utilities

Single Responsibility Principle:
Amagar un element, afegir un marge, centrar un text...

Immutabilitat

Dissenyades per **excepcions**

Tenen molta especificitat:
!important

▲ 7-Utilities

🔗 `_main.scss`

🔗 `_utilities.clearfix.scss`

🔗 `_utilities.hidden.scss`

🔗 `_utilities.print.scss`

🔗 `_utilities.spacing.scss`

7 - Utilities

```
.hidden {  
  display: none !important; }
```

```
.color-primary {  
  color: $color-primary !important; }
```

```
.width-1/2 {  
  width: 50% !important; }
```

En resum:

Abast
ampli



Abast
limitats

Selectors
universals

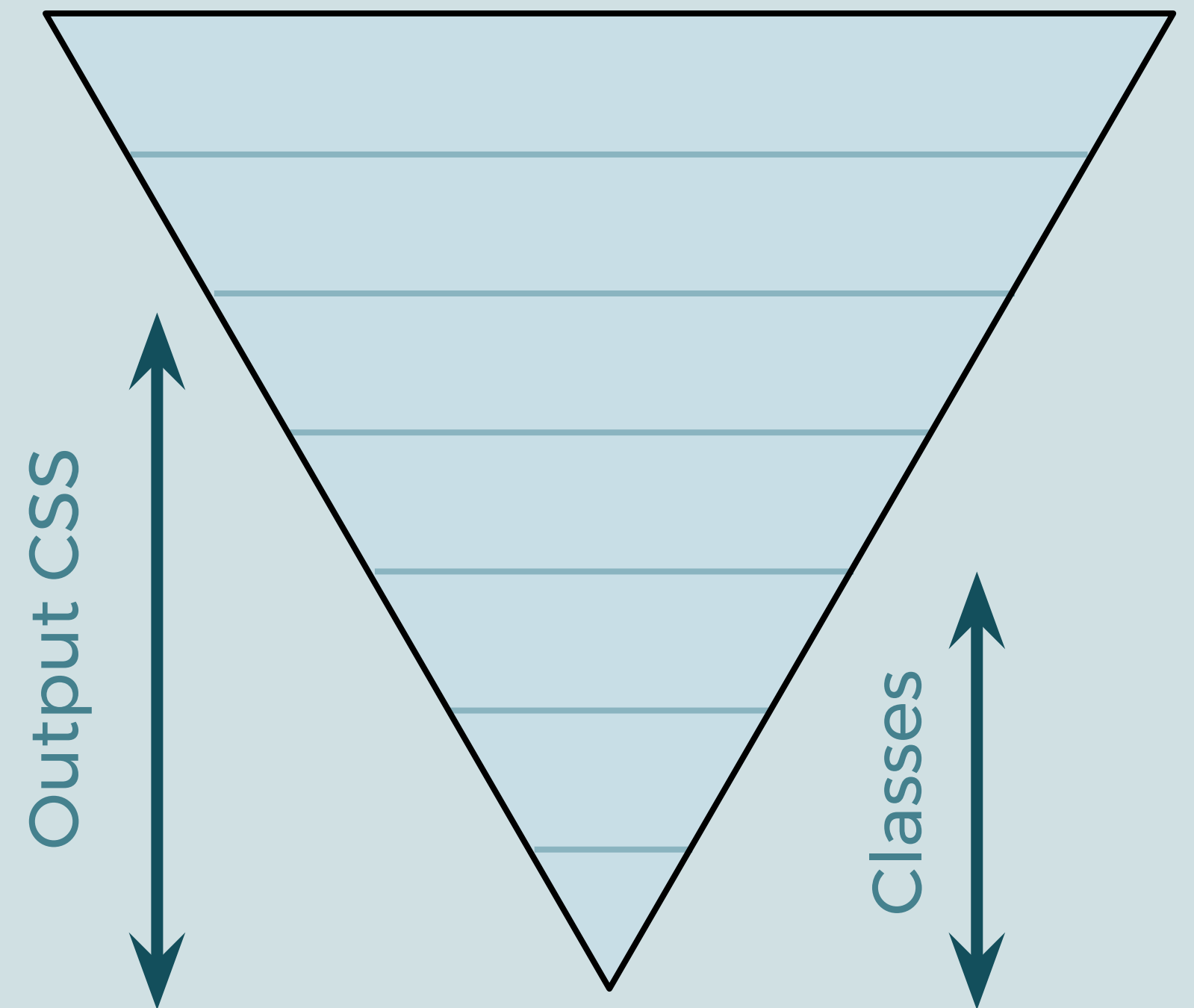


Selectors
específics

Baixa
especificitat



Alta
especificitat



La majoria d'elements de la UI estan formats d'Objects i Components

Com sabem
quina classe és
un Object i quina
un Component?

```
<div class="media testimonial">
```

```
...
```

```
</div>
```

```
<div class="media featured-tweet">
```

```
...
```

```
</div>
```

Introducing...

Namespaces

Namespaces

Indicador (prefix) per saber a quina capa pertany cada classe.

```
<div class="media testimonial">
```

```
...
```

```
</div>
```

```
<div class="media featured-tweet">
```

```
...
```

```
</div>
```

Namespaces

Indicador (prefix) per saber a quina capa pertany cada classe.

```
<div class="o-media c-testimonial">
```

```
...
```

```
</div>
```

```
<div class="o-media c-featured-tweet">
```

```
...
```

```
</div>
```

Namespaces

.o- per Objects

.c- per Components

.u- per Utilities

.js- per javascript hooks



Les 3 últimos capes del triangle!

Separation of Concerns
(SoC)

Namespaces

.o- per Objects

.c- per Components

.u- per Utilities

.js- per javascript hooks

.o-pack

.c-button

.u-text-right

.js-featured-items

Però...

“Només hi ha dues coses complicades en l'enginyeria del software: la invalidació de la caché i **posar nom a les coses.**”

- *Phil Karlton*

Exercici: esborrar totes les classes relacionades amb l'usuari per poder reaprofitar el codi.

```
<div class="media user premium">  
  <img src="" alt="" class="img photo avatar" />  
  <p class="body bio">...</p>  
</div>
```

Ets un exagerat i t'ho estàs
fent venir bé.


```
<div class="border logged text-normal">  
  <span class="name">...</span>  
  <span class="role">...</span>  
</div>
```

```
<li class="border dropdown text-normal">  
  <a class="adminLink">  
    <span class="adminMenu">...</span>  
  </a>  
</li>
```

Introducing...

Nomenclatura BEM

Block - Element - Modifier

Nomenclatura BEM

Block: L'arrel del component/object. El contenidor.

Element: Un element intern del Block.

Modifier: Una variant / extensió d'un Block o un Element.

Nomenclatura BEM

`.block__element--modifier`

```
.search-box {}
```

```
.search-box__input {}
```

```
.search-box--christmas {}
```

```
.search-box__input--rounded {}
```



Nomenclatura BEM - Media Object

```
.media {  
  display: flex;  
  align-items: flex-start; }
```

```
.media .body {  
  flex: 1; }
```

```
.media .image {  
  margin-right: 1em; }
```

```
<div class="media">  
  <div class="image">...</div>  
  <div class="body">...</div>  
</div>
```

Nomenclatura BEM - Media Object

```
.media {  
  display: flex;  
  align-items: flex-start; }
```

```
.media__body {  
  flex: 1; }
```

```
.media__image {  
  margin-right: 1em; }
```

```
<div class="media">  
  <div class="media__image">...</div>  
  <div class="media__body">...</div>  
</div>
```


Nomenclatura BEM - Media Object

```
.o-media {  
  display: flex;  
  align-items: flex-start; }
```

```
.o-media__body {  
  flex: 1; }
```

```
.o-media__image {  
  margin-right: 1em; }
```

```
<div class="o-media">  
  <div class="o-media__image">...</div>  
  <div class="o-media__body">...</div>  
</div>
```

Nomenclatura BEM - Media Object

```
.o-media--large {  
  > .o-media__image {  
    margin-right: 2em;  
  }  
}
```

```
<div class="o-media o-media--large">  
  <div class="o-media__image">...</div>  
  <div class="o-media__body">...</div>  
</div>
```

Nomenclatura BEM - Example

// GOOD

```
.person {}
```

```
.person__head {}
```

```
.person__face {}
```

```
.person__eye {}
```

```
.person--female {}
```

```
.person__eye--blue {}
```

// BAD

```
.person__head__face {}
```

```
.person__head__face__eye {}
```

```
.person--female__head {}
```

Nomenclatura BEM - Exemple (2)

```
// GOOD(?)
```

```
.page { }
```

```
.content { }
```

```
.sub-content { }
```

```
.footer { }
```

```
.footer__copyright { }
```

```
// BAD
```

```
.page__sub-content { }
```

```
.page__footer { }
```

```
.page__copyright { }
```

Les classes no han de reflectir l'esquelet del DOM.

```
.btn {  
  font-size: 1.2em;  
}
```

```
.promo .btn {  
  font-size: 2em;  
}
```

Una classe,
dos outputs.

No ens agrada.

```
.btn {  
  font-size: 1.2em;  
}
```

```
.btn--large {  
  font-size: 2em;  
}
```

Immutable
Menys específic
Reaprofitable
Simple

Nomenclatura BEM - Exemple (4)

```
.btnLogin {  
  display: inline-block; ← Base  
  padding: 20px; ← Estrucutural  
  background-color: #BADA55; ← Cosmètic  
  color: white; ← Cosmètic  
}
```

Nomenclatura BEM - Exemple (4)

```
.btn {  
  display: inline-block; }  
  
.btn--large {  
  padding: 20px; }  
  
.btn--positive {  
  background-color: #BADA55;  
  color: white; }
```

```
.btn--small {}  
  
.btn--disabled {}  
  
.btn--secondary {}  
  
.btn--full {}  
  
.btn--negative {}  
  
...
```


“The most reusable components are
those with class names that are
independent of the content”

- *Nicholas Gallager*

Exercici: esborrar totes les classes relacionades amb l'usuari per poder reaprofitar el codi.

```
<div class="media user premium">  
  <img src="" alt="" class="img photo avatar" />  
  <p class="body bio">...</p>  
</div>
```

Exercici: esborrar totes les classes relacionades amb l'usuari per poder reaprofitar el codi.

```
<div class="o-media c-user c-user--premium">  
  <img src="" alt="" class="o-media__img c-user__photo c-avatar" />  
  <p class="o-media__body c-user__bio">...</p>  
</div>
```

Si hem aplicat BEM i Namespaces...

Exercici: esborrar totes les classes relacionades amb l'usuari per poder reaprofitar el codi.

```
<div class="o-media c-user c-user--premium">  
  <img src="" alt="" class="o-media__img c-user__photo c-avatar" />  
  <p class="o-media__body c-user__bio">...</p>  
</div>
```

1 Object (o-media)

Exercici: esborrar totes les classes relacionades amb l'usuari per poder reaprofitar el codi.

```
<div class="o-media c-user c-user--premium">  
  <img src="" alt="" class="o-media__img c-user__photo c-avatar" />  
  <p class="o-media__body c-user__bio">...</p>  
</div>
```

1 Object (o-media)

1 Component (c-avatar)

Exercici: esborrar totes les classes relacionades amb l'usuari per poder reaprofitar el codi.

```
<div class="o-media c-user c-user--premium">  
  <img src="" alt="" class="o-media__img c-user__photo c-avatar" />  
  <p class="o-media__body c-user__bio">...</p>  
</div>
```

1 Object (o-media)

5-Objects/_**objects.media.scss**

1 Component (c-avatar)

6-Components/_**components.avatar.scss**

1 Component (c-user)

6-Components/_**components.user.scss**

wow__such--amaze

¿xque?
¿xque?

xque tantas clases?

jo posaré !important
i al palco...

xque no
me kiere?

XQQ?

AFONCU Y U DO DIS
bete en bici

wow much block
element

XQQ?

“Tot això t’ho has tret de la màniga”


```
▼ <div class="badgeList m-twoRows lazyLoadingList">
  ▼ <ul class="lazyLoadingList__list sc-list-nostyle sc-clearfix">
    ::before
    ► <li class="badgeList__item">...</li>
    ► <li class="badgeList__item">...</li>
    ► <li class="badgeList__item">...</li>
    ► <li class="badgeList__item">...</li>
```

SoundCloud

```
▼ <div class="section__content">
  ▶ <div class="section__icon ak-in-view--start ak-in-view--end">...</div>
    <div class="section__title">
      Unlimited storage for all your photos and videos.
    </div>
  ▶ <div class="section__body">...</div>
</div>
<div class="section__image">
```

Google Pixel

```
▼ <section class="item-list clearfix">
  ▼ <ol class="item-order itemlist hotellist clearfix">
    ▼ <li class="hotel item-order__list-item js_co_item">
      ▼ <article class="item bg-white">
        ▶ <div>...</div>
        ▶ <div class="item__image-area">...</div>
        ▶ <div class="item__flex-column">...</div>
        ▶ <div class="item__slideout">...</div>
      </article>
    </li>
```

Trivago

```
▼<div class="top-stories__freeform-twocolumns o-teaser-
collection o-grid-row js-track-scroll-event">
  ▼<div>
    ▼<div class="o-teaser o-teaser--small o-teaser--has-image
o-teaser--article o-teaser--syndicatable" data-o-component=
"o-teaser">
      ▼<div class="o-teaser__content">
        ▶<div class="o-teaser__meta">...</div>
        ▶<div class="o-teaser__heading js-teaser-heading">...
</div>
        ▶<p class="o-teaser__standfirst">...</p>
        ▶<div class="o-teaser__timestamp">...</div>
```

Financial Times

```
▼ <div class="slider__item">
  ▼ <article class="post post--inspiration">
    ▶ <style>...</style>
    ▼ <header class="post__header post__header--8032 js-slide">
      ▼ <div class="post__background post__background--default"
        ::after
        </div>
        <div class="post__background post__background--mobile">
      ▼ <div class="post__heading">
        ▶ <div class="post__heading__content">...</div>
        ▶ <div class="post__summary">...</div>
        </div>
```

Typeform

```
▼ <div class="o-page__content u-flex-center">
  <h1 class="u-text-uppercase">Category 2</h1>
  ▼ <div class="o-list-flex u-margin-top-large c-experience-list">
    <a href="/experience/1" class="o-list-flex__item c-button c-button--white
    c-button--ghost c-experience-list__item c-experience-list__item--1">
      Barcelona 360º 1
    </a>
    <a href="/experience/2" class="o-list-flex__item c-button c-button--white
    c-button--ghost c-experience-list__item c-experience-list__item--2">
      Barcelona 360º 2
    </a>
  ▶ <div class="c-button c-button--disabled">...</div>
```

We Barcelona

“If you’re looking for a **good way to organize** your CSS, BEM is a really good starting point, both from a **structure** point-of-view, but also because of the **simplifications** of style lookup.”

- Google Web Developers (<https://goo.gl/c7U4JM>)

Bones pratiques

Show CSS some love

Bones pràctiques

1. No utilitzar #IDs
 2. Inception Rule
 3. Encapsular frameworks de tercers
 4. CSS linting
- ...

1. No utilitzar #IDs

...als fulls d'estil!

1. No utilitzar #IDs

...als fulls d'estil!

```
#loginForm {  
  al: palco;  
}
```

Res supera l'especificitat d'un estil definit en un #id

Impossible de reaprofitar

Simplifica el codi: com si no existíssin

2. Inception Rule



2. Inception Rule

No arribar mai als 4 nivells de profunditat

*...I AMB MENYS
N'HI HA PROU!!*

Regla d'or:

Sempre que declaris un estil, fes-ho amb la **menor quantitat** de selectors possible.

```
body {  
  div.wrapper {  
    article.featured {  
      > p:first-of-type {  
        font-size: 1.2em;  
      }  
    }  
  }  
}
```

Regla d'or:

Sempre que declaris un estil, fes-ho amb la **menor quantitat** de selectors possible.

```
article#Featured {  
  > p.introText {  
    font-size: 1.2em;  
  }  
}
```


Regla d'or:

Sempre que declaris un estil, fes-ho amb la **menor quantitat** de selectors possible.

```
.c-featured-news__intro {  
    font-size: 1.2em;  
}
```

Regla d'or:

Sempre que declaris un estil, fes-ho amb la **menor quantitat** de selectors possible.

4 nivells
(tot i que només
en compten 3)

```
.block {  
  .block--mod {  
    .block__element {  
      &:hover {  
        font-size: 1.2em;  
      }  
    }  
  }  
}
```

```
2369                                     input {
2370                                         display: inline-block;
2371                                     }
2372                                 }
2373                            }
2374                       }
2375                   }
2376               }
2377           }
2378       }
2379
2380       .map {
2381           width: 100%;
2382           height: 300px;
2383       }
2384   }
2385 }
2386 }
2387 }
2388 }
2389
```

xD

```
body.product .content-body-wrapper.tabs-block
.switch .offer section.accommodation .content-width
.accommodation .select-accommodation-box .list .map
{ width: 100%; height: 300px; }
```

```
body.product .content-body-wrapper.tabs-block
.switch .offer section.accommodation .content-width
.accommodation .select-accommodation-box .list
.list-content .items .accommodation-info
#formAccomodation .accommodation-form .mealplan
.select-price input { display: inline-block }
```

3. Encapsular frameworks

3. Encapsular frameworks

```
<p class="o-media c-user c-user--premium [ uk-grid uk-width-1-3 ]">
```

```
<p class="uk-grid o-media c-user c-user--premium uk-width-1-1">
```

Realment necessites un altre framework?

4. (S)CSS Linting

`scss/7-Utilities/_utilities.hidden.scss`

<code>10:1</code>	<code>✘</code>	<code>Expected indentation of 4 spaces</code>	<code>indentation</code>
<code>10:9</code>	<code>✘</code>	<code>Expected single space after ":" with a single-line declaration</code>	<code>declaration-colon-space-after</code>
<code>11:1</code>	<code>✘</code>	<code>Expected indentation of 4 spaces</code>	<code>indentation</code>
<code>11:7</code>	<code>✘</code>	<code>Unexpected named color "blue"</code>	<code>color-named</code>
<code>11:7</code>	<code>✘</code>	<code>Expected single space after ":" with a single-line declaration</code>	<code>declaration-colon-space-after</code>
<code>12:1</code>	<code>✘</code>	<code>Expected indentation of 4 spaces</code>	<code>indentation</code>
<code>12:5</code>	<code>✘</code>	<code>Expected single space before "{"</code>	<code>block-opening-brace-space-before</code>
<code>12:8</code>	<code>✘</code>	<code>Expected double colon pseudo-element notation</code>	<code>selector-pseudo-element-colon-notation</code>
<code>12:13</code>	<code>✘</code>	<code>Expected single space before "{"</code>	<code>block-opening-brace-space-before</code>
<code>12:21</code>	<code>✘</code>	<code>Expected single space after ":" with a single-line declaration</code>	<code>declaration-colon-space-after</code>
<code>12:26</code>	<code>✘</code>	<code>Expected single space after "," in a single-line function</code>	<code>function-comma-space-after</code>
<code>12:28</code>	<code>✘</code>	<code>Expected single space after "," in a single-line function</code>	<code>function-comma-space-after</code>

Recap

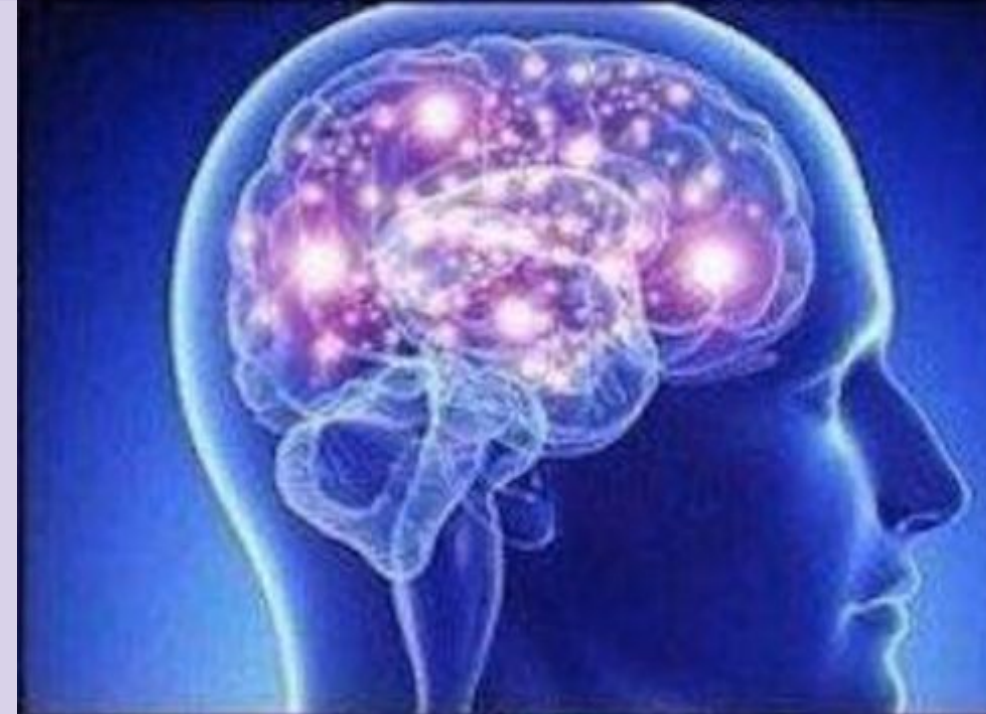
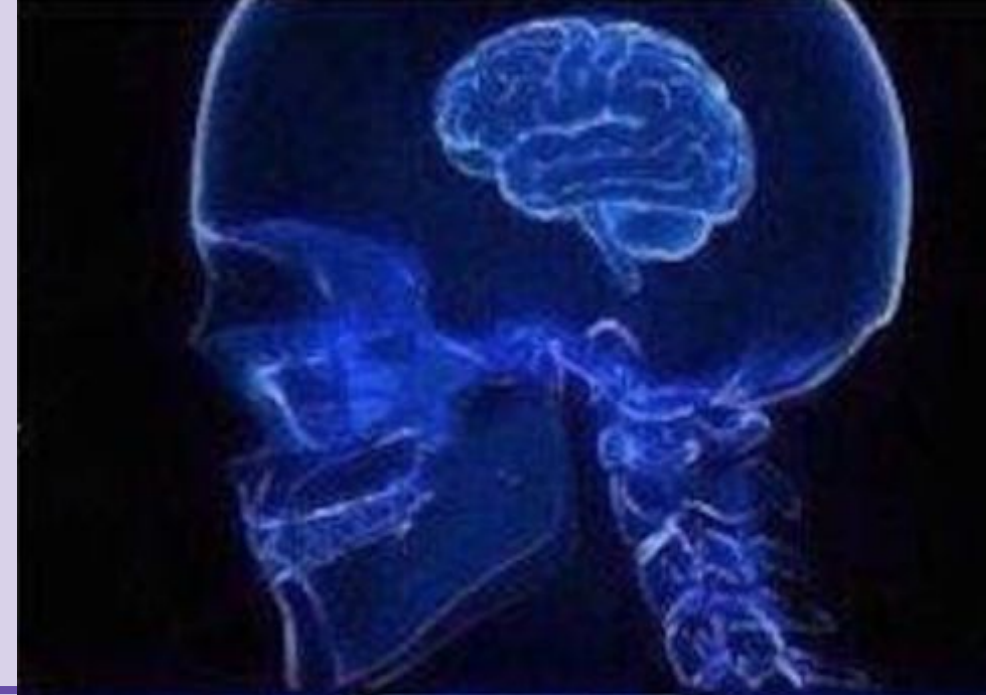
1. **Estructura ITCSS:** 7 capes
2. **Nomenclatura BEM:** block__element--mod
3. **Namespacing:** prefix de capa
4. **Bones pràctiques:** IDs, linting, inception rule...

Sass

Sass + ITCSS

**Sass + ITCSS +
Namespaces**

**Sass + Namespaces +
ITCSS + BEM**



Què hem aconseguit amb tot això?

1. Estructura de selectors plana (*no nesting*)
2. Codi organitzat (*ja no escrivim estils al final*)
3. Les classes fan una sola cosa, i la fan bé
4. Desacoplem per reaprofitar



Normandy CSS

Documentació i Boilerplate

Normandy Vue

Implementació d'Objects en VueJS

...i a partir d'aquí...?

Adoptar la metodologia en nous projectes.

Aplicar-ne parts a projectes existents

(Capa de Settings i Tools? BEM? Bones pràctiques?

Encapsular BEM - noBEM?)

...i a partir d'aquí...?

Ampliar la guia!

...media queries?

...més Objects (Grid, Layout, Island, Bare List....)?

...més namespaces (.is-, .has-)?

...modificar/afegir regles linting?

...magic numbers?

CSS
IS
AWESOME



NORMANDY

IS

AWESOMER

Thanks!

Dubtes?

(segur que sí!)

Demo time!