PAMPLONA SOFTWARE CRAFTERS











cabify Codium aute<u>n</u>tia

eventbrite





#SCPNA

HELLO!



DIANA ACEVES

DEVELOPER EN topi

AKA "PUCELANA BORDE FEMINAZI"

@diana_aceves_

LAS PASTILLAS ROJAS DEL CSS

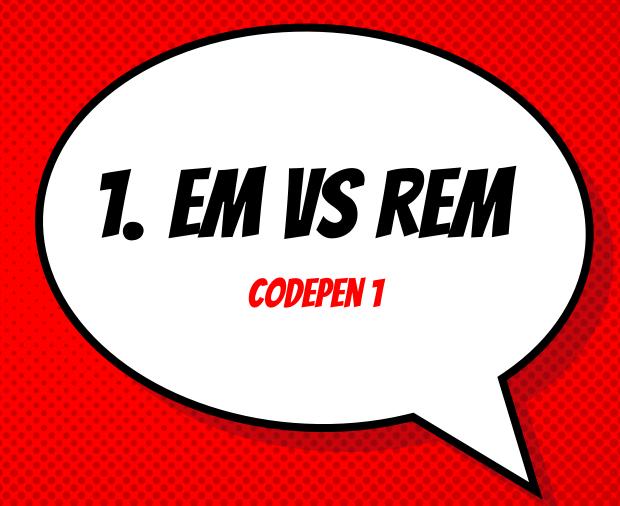


PASTILLAS ROJAS

- 1. EM US REM
- 2. CASCADA
- 3. ESPECIFICIDAD
- 4. POSITION
- 5. FLEX
- 6. MÁS COSITAS...

COLECCIÓN CODEPEN

- × Tenéis que "forkear" los pens al abrirlos.
- × URL colección: http://tiny.cc/scpna2022
- × URL repo:
 https://github.com/dianaaceves/scpna2022

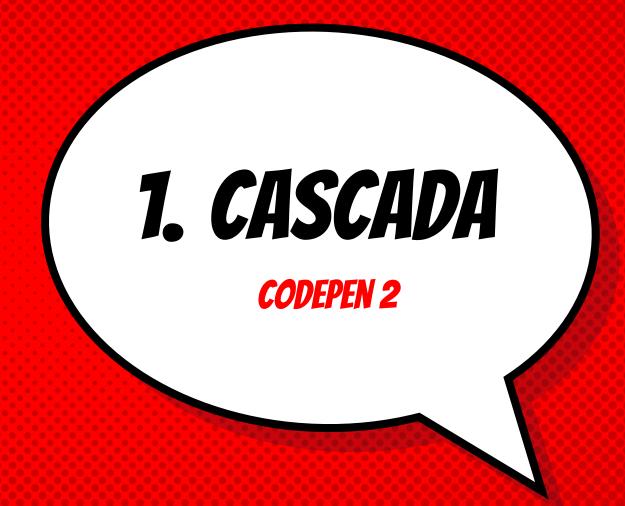


rem

- x ACCESIBILIDAD: Responde a los cambios de tamaño del usuario.
- x Tiene como referencia el font-size del elemento **root** (el html).
- x Es el **mismo** para todo el documento.

em

- x Tiene como referencia el font-size del elemento actual o heredado de un ancestro que lo haya establecido.
- x Cambia cuando haya cambiado el font-size.



ENCUESTA TWITTER



How well do you know CSS? 🦃

Given these classes:

```
.red {
  color: red;
}
.blue {
  color: blue;
}
```

Which color would these divs be?

```
<div class="red blue"> <div class="blue red">
```

Traducir Tweet

- A) First red, second blue
- B) First blue, second red
- C) Both blue
- D) Both red

9% First red, second blue44% First blue, second red43% Both blue4% Both red

14,517 votos · Resultados finales

CASCADA

- **x** CSS: Cascading Style Sheets.
- x Se encarga de resolver los conflictos que surgen cuando diferentes hojas de estilo, reglas y propiedades CSS se aplican al mismo elemento.

Nos indica QUIÉN GANA.

CASCADA

La cascada es un algoritmo que decide en función de:

- X Hoja de estilos a la que pertenece: author (nosotros) > user > user agent (default navegador)
- Orden en el código: (*)
 En igualdad de condiciones, gana lo que está escrito después

EN EL CSS.

× Especificidad: Gana el selector más específico.



CASO 1: MISMO ELEMENTO ATACADO POR DISTINTOS SELECTORES

- × Elemento y pseudo-elemento
- × Clase, pseudo-clase y atributo
- × ID
- × Estilos en línea
- × !important
- × !important en línea



- × Elemento y pseudo-elemento
 div, h1, ::before, ::first-letter, ::first-line
- x Clase, pseudo-clase y atributo
 .red, :first-child, :not(), :hover, [href^=mailto]
- × ID: #menu
- x Estilos en línea: <div style="color: red;">

CODEPEN: 3

PERO ¿QUÉ PASA CUANDO ME ENCUENTRO ESTO?

```
#js-main-menu a{}
.menu--main ul li.menu__item a.menu__link{}
#js-main-menu .menu__item:nth-child(2) a{}
```

PERO ¿QUÉ PASA CUANDO ME ENCUENTRO ESTO?

```
#js-main-menu a{}
.menu--main ul li.menu__item a.menu__link{}
#js-main-menu .menu__item:nth-child(2) a{}
```



CASO 2: COMBINACIONES DE SELECTORES

REGLA DEL 0 0 0 0

- Unidades -> elemento y pseudo-elemento
- Decenas -> clase, pseudo-clase, atributo
- × Centenas -> ID
- Unidades de millar -> estilos en línea

(Comparamos resultados y vemos cuál es mayor)

CODEPEN 4

COSAS A TENER EN CUENTA CON !IMPORTANT

- × Se carga los estilos que vienen por JS
- Me obliga a <u>arrastrarlo por todas las media queries</u>
- NO USAR, salvo para:
 - Sobreescribir otros !important
 - v Utilities -> is-hidden

VAMOS A MOVER COSAS

VAMOS A MOVER COSAS PERO VAMOS A MOVER COSAS...

VAMOS A MOVER COSAS PERO VAMOS A MOVER COSAS... QUERIENDO

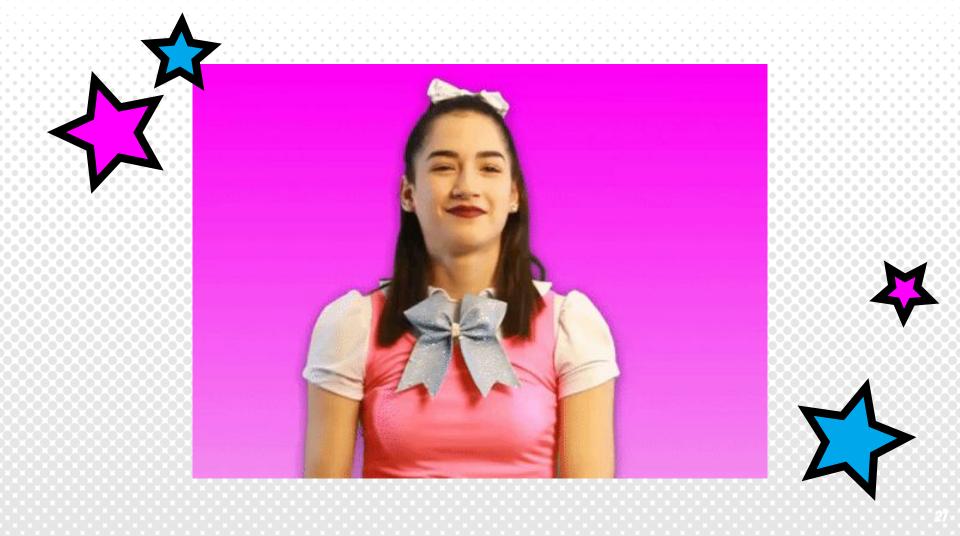




PROGRAMADORES

POSITION









POSITION

- × **STATIC (DEFAULT):** se coloca donde le toca según el flujo HTML.
- × RELATIVE: se coloca respecto al sitio donde está y el resto NO ocupan su sitio.
- * **ABSOLUTE:** se coloca <u>respecto al 0,0 de su primer ancestro que</u> <u>tenga position != static y el resto SÍ ocupan su sitio</u>.
- × **FIXED:** se coloca r<u>especto al 0,0 del viewport</u> y <u>el resto SÍ ocupan</u> <u>su sitio.</u>



¿TENGO QUE PONER COSITAS UNAS AL LADO DE OTRAS?

FLEXBOX

ETENGO QUE CENTRAR COSITAS EN VERTICAL O/Y HORIZONTAL?

FLEXBOX

CODEPEN 6

THAS.