!Hola!

Soy Diana Aceves

Senior Developer en **DEISER**

Twitter: **@diana_aceves_**

PLAN DE TRABAJO

1. **Fundamentos CSS** (3h.)
2. **Personalización de frameworks** (1h.)
3. **Web crafter** (3h.)

Juguemos a KAHOOT

INFO

1. Colección CODEPEN:

https://tinyurl.com/taller-eurohelp

Tenéis que FORKear los pens cuando los abráis.

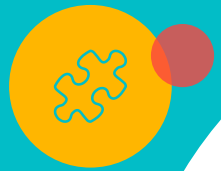
2. Actualizad repositorio.

BLOQUE I

Fundamentos CSS

Fundamentos CSS

- Cascada
- Especificidad
- Box model
- Display
- Position



1. Cascada

Pen 1: ¿De qué color es?

1. Cascada

- **CSS: Cascading** Style Sheets.
- La cascada se refiere al **ORDEN en el que el navegador aplica las reglas CSS.**
- En igualdad de condiciones, **gana lo que está escrito después en el CSS.**

Encuesta Twitter a esta pregunta

Max Stoiber
@mxstbr

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

Más de 8000 personas contestan mal y ¡¡gana una respuesta incorrecta!!

2. Especificidad

2. Especificidad

Caso 1: mismo elemento atacado con **distintos selectores**.
En igualdad de condiciones, **gana el selector más específico.**

Especificidad creciente

- Tipo (elemento) y pseudo-elemento
- Clase, pseudo-clase y atributo
- ID
- Estilos en línea
- !important
- !important en línea

\- ESPECIFICIDAD
\- **FUERZA**

\+ ESPECIFICIDAD
\+ **FUERZA**

2. Especificidad

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

2. Especificidad

¿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{}
```

2. Especificidad

¿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{}
```

!important

2. Especificidad

Caso 2: mismo elemento atacado por **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)

2. Especificidad - ¿Quién gana?

1. `#js-main-menu a{}`
   `.menu--main ul li.menu__item a.menu__link{}`

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

Pen 2: Especificidad

2. Especificidad

Cosas a tener en cuenta con !important

- **Se carga los estilos que vienen por JS** (luego que se rompen cosas…).
- Me obliga a arrastrar el **!important** en todas las **media queries.**
- No usar NUNCA salvo:
  - para **sobreescribir otros !importants** -> truqui para añadir especificidad: body, html, duplicar.
  - para **utilities** -> .is-hidden, .is-warning...

Hemos pintado y coloreado.

Hemos pintado y coloreado.

Vamos a empezar a mover cosas.

Hemos pintado y coloreado.

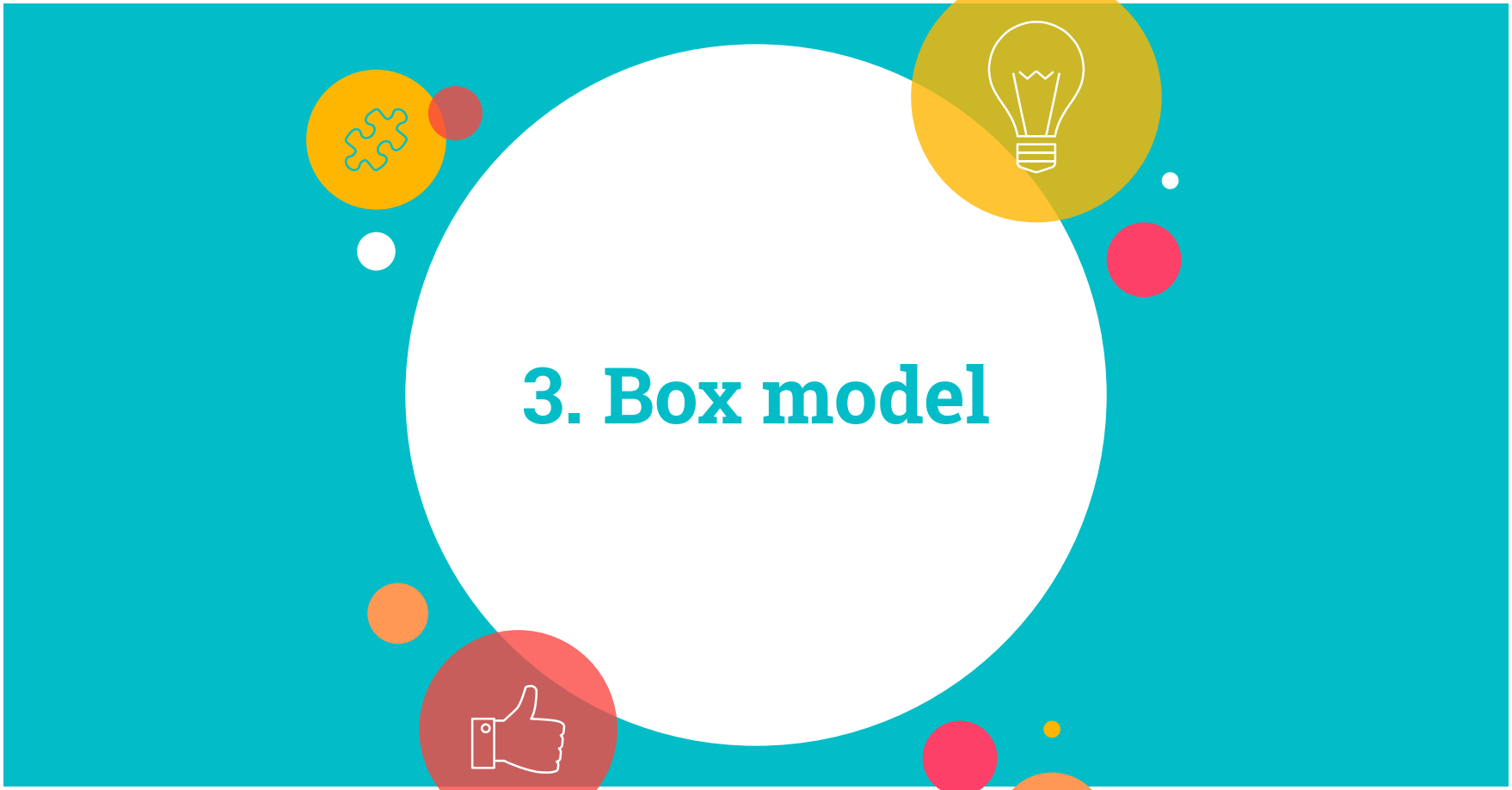
Vamos a empezar a mover cosas.

Pero vamos a mover cosas...

Hemos pintado y coloreado.

Vamos a empezar a mover cosas.

Pero vamos a mover cosas...QUERIENDO.


3. Box model

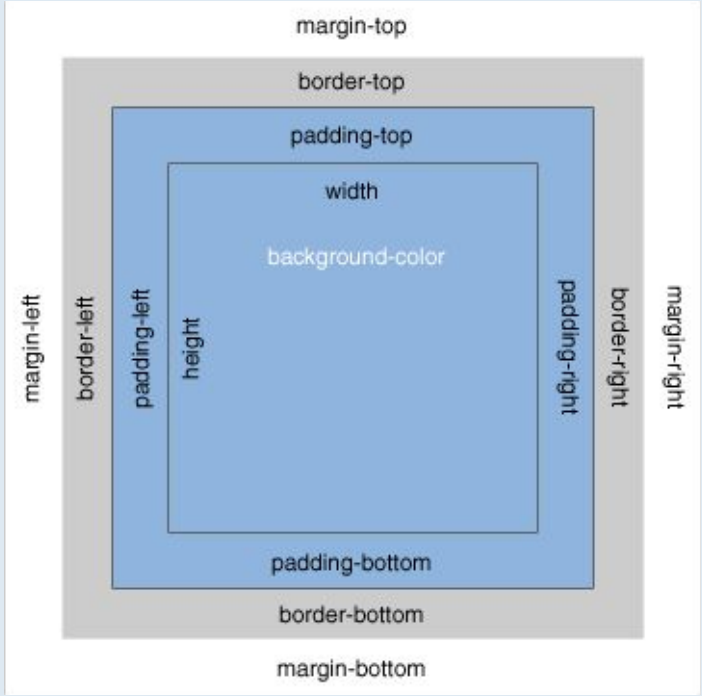
3. Box model

El **box model** describe la caja que se forma alrededor de un elemento.

Es **“lo que ocupa”**.

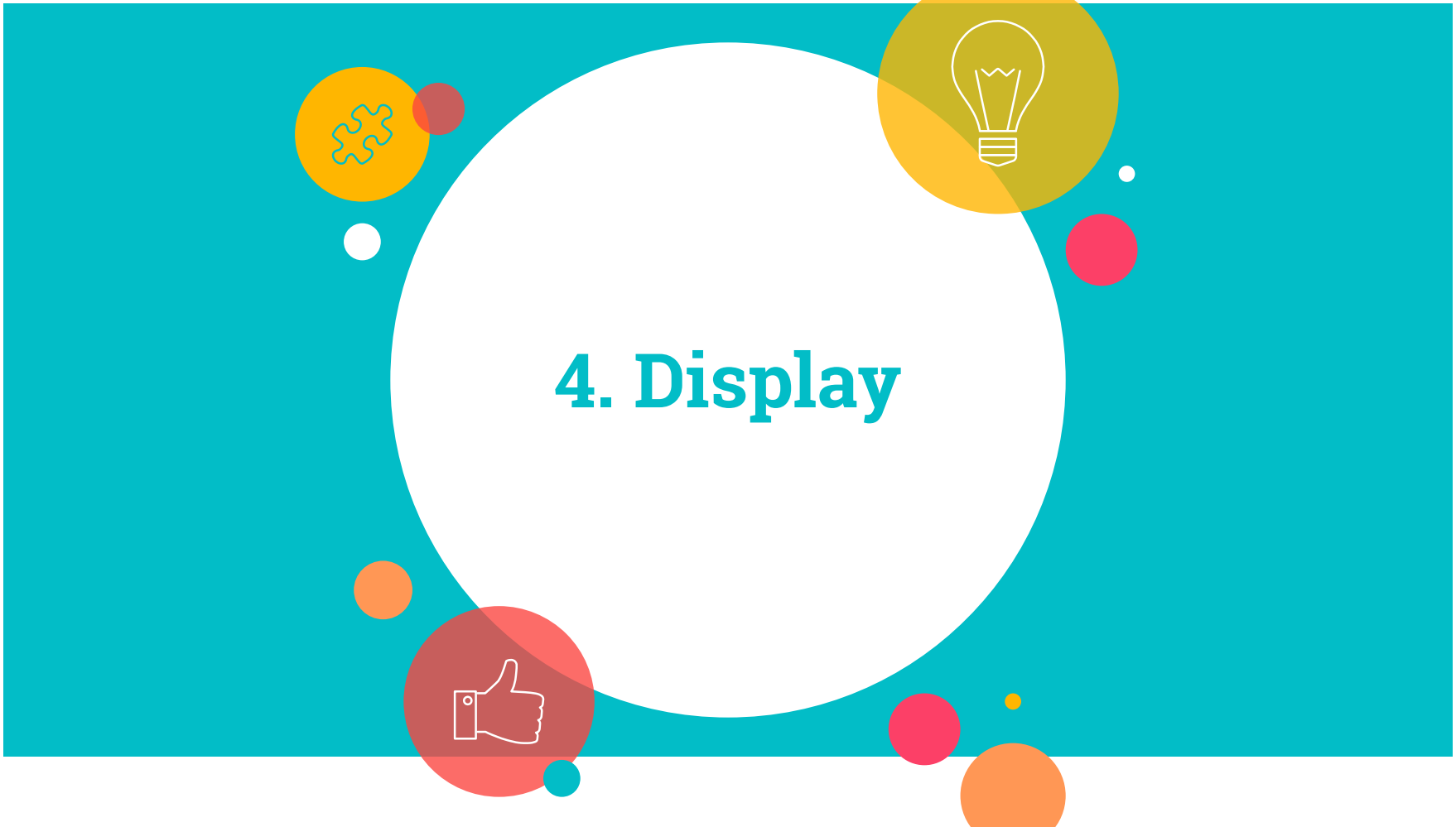
Pen 3: Box Model

3. Box model



¡NO OS OLVIDÉIS!

box-sizing: border-box;

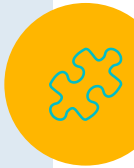

4. Display

Display

Determina cómo se visualiza un elemento y cómo se comporta con los elementos de su alrededor

none	table	**flex**
inline	inline-table	inline-flex
block	table-cell	**grid**
inline-block	table-column	inline-grid
list-item	table-row	...y más.
inline-list-item	table-caption	

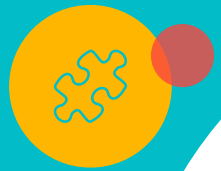
Pen 4: Display

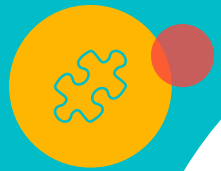


4. Display

	Set height/width	Margin top/bottom	Line break after
Inline	NO	NO	NO
Block	YES	YES	YES
Inline-block	YES	YES	NO

¿Y si jugáramos ahora a KAHOOT?





5. Position

Citas célebres

“Que alguien ponga música porque se nos va a mover todo”

Diana

Pen 5: Position (VER CAPTURAS)

5. Position

- **Static (default):** se coloca donde le toca según el flujo HTML.
- **Relative:** se mueve **respecto al sitio donde está** y el resto **NO** ocupan su espacio.
- **Absolute**: se coloca **respecto al 0,0 de su primer ancestro que tenga position != static** y el resto **SÍ** ocupan su sitio. (“Se me mueve todo”).
- **Fixed**: se coloca **respecto al 0,0 del viewport** y el resto **SÍ** ocupan su sitio.
- **Z-index**: orden en la pila de contenidos. **Solo funciona si hay position != static.** (Como vea un z-index 9999 os reviento).

BLOQUE II

Personalización de Frameworks

BLOQUE III

Crafter web responsive

Referencias

- BEM by example
- Emmet
- SASS basics
- em vs rem
- 0to255 color variations tool
- calc()
- Viewport units: vh y vw
- Flexbox Guide - CSS Tricks
- Grid Layout Guide - CSS Tricks
- Rock’n’Grid - Diana Aceves