

# Fundamentos de la web

# Esquema básico web

- ▶ Cliente/Servidor
- ▶ Navegador y Servidor hablan http
- ▶ Cliente pide una **url** y el servidor le devuelve el **archivo** correspondiente
  - ▶ html
  - ▶ css
  - ▶ js
  - ▶ json, xml
  - ▶ jpg, png, gif, pdf, mpeg ... (lo que sea)

- ▶ ¿Dónde está php, java, python o ruby?

- ¿Dónde está php, java, python o ruby?
- ¿Dónde se ejecuta javascript?

# HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <h1>Page Title</h1>
    <p>This is a really interesting paragraph.</p>
  </body>
</html>
```

Muchos Elementos

Estructura y contenido. **NO estilo**

Atributos:

```
<a href="http://d3js.org/">The D3 website</a>
```

Clases:

```
<p class="importante"> Se termina la corrupción </p>
```

Ids:

```
<div id="contenido"></div>
```

Comentarios:

```
<!-- Esto está comentado -->
```

# DOM

Jerarquía (o árbol) con padres, hijos, vecinos, ascendientes y descendientes.

```
<html>
  <body>
    <h1>Breaking News</h1>
    <p></p>
  </body>
</html>
```

**Abre la herramienta de desarrollo de tu navegador: F12**

# Modelo de cajas

- ▶ Todo está en cajas.
- ▶ Los principales: **div** y **span**  
`display:block` y `display:inline`
- ▶ Margin, Border y Padding



# CSS

```
body {  
    background-color: black;  
    color: white;  
}
```

Siempre esto:

```
selector[, selector2, selector3, ...] {  
    propiedad: valor;  
    propiedad: valor;  
    propiedad: valor;  
    propiedad: valor;  
}
```

# Selectores

Por etiqueta

```
div /* Selecciona todos los divs
```

Por clase

```
.importante /* Selecciona todo lo que tenga clase "importante"  
div.importante /* Sólo div con clase "importante"  
.uno.dos /* Con varias clases
```

Por ID

```
#contenido /* El único elemento con id "contenido"  
#contenido.seleccionado /* Sólo cuando tiene esa clase
```



# Estilos

- ▶ Una lista enorme de propiedades
- ▶ Color:
  - ▶ Colores con nombre: orange
  - ▶ Hex: #3388aa o #38a
  - ▶ RGB: rgb(10, 150, 20)
  - ▶ RGB con alpha: rgba(10, 150, 20, 0.5)
- ▶ Medidas:
  - ▶ 630px
  - ▶ 23m
  - ▶ 45%

# ¿Dónde poner los estilos?

- ▶ Nunca:

```
<p style="color: blue; font-size: 48px;  
      font-style: italic;">No hacer</p>
```

- ▶ Embebidos en el **head**:

```
<style type="text/css">  
  div {  
    background-color: red;  
    font-size: 24px;  
    font-weight: bold;  
  }  
</style>
```

# ¿Dónde poner los estilos?

- ▶ Mejor en su archivo:

```
<head>  
  <link rel="stylesheet" href="style.css">  
</head>
```

- ▶ Se aplican en **cascada**:

```
<head>  
  <link rel="stylesheet" href="style.css">  
  <link rel="stylesheet" href="sobrescribe.css">  
</head>
```

# Javascript

- ▶ Interpretado
  - ▶ Dinámico
  - ▶ NO tipado
- 
- ▶ Mala fama... medio merecida
  - ▶ Súper optimizado

# ¿Dónde poner el código javascript?

- ▶ Embebido en el **body**:

```
<script type="text/javascript">  
    alert("Hello, world!");  
</script>
```

- ▶ Mejor en su archivo (head o body):

```
<script rel="text/javascript" href="micodgio.js">  
</script>
```

## Tipos

- ▶ Variables



# Tipos

- ▶ Variables
- ▶ Strings

# Tipos

- ▶ Variables
- ▶ Strings
- ▶ Arrays

# Tipos

- ▶ Variables
- ▶ Strings
- ▶ Arrays
- ▶ Objectos

# Tipos

- ▶ Variables
- ▶ Strings
- ▶ Arrays
- ▶ Objetos
- ▶ Funciones

# Tipos

- ▶ Variables
- ▶ Strings
- ▶ Arrays
- ▶ Objetos
- ▶ Funciones
  - ▶ lambdas y nombradas

# Tipos

- ▶ Variables
- ▶ Strings
- ▶ Arrays
- ▶ Objectos
- ▶ Funciones
  - ▶ lambdas y nombradas
  - ▶ clousures

# Tipos

- ▶ Variables
- ▶ Strings
- ▶ Arrays
- ▶ Objectos
- ▶ Funciones
  - ▶ lambdas y nombradas
  - ▶ clousures
  - ▶ callbacks

# Tipos

- ▶ Variables
- ▶ Strings
- ▶ Arrays
- ▶ Objectos
- ▶ Funciones
  - ▶ lambdas y nombradas
  - ▶ clousures
  - ▶ callbacks
- ▶ JSON



# Ojo, fallos comunes

- ▶ No se define el tipo, una variable puede cambiar su tipo sin más

# Ojo, fallos comunes

- ▶ No se define el tipo, una variable puede cambiar su tipo sin más
- ▶ Elevado de la declaración de variables

# Ojo, fallos comunes

- ▶ No se define el tipo, una variable puede cambiar su tipo sin más
- ▶ Elevado de la declaración de variables
- ▶ El ámbito de las variables es la función

# Ojo, fallos comunes

- ▶ No se define el tipo, una variable puede cambiar su tipo sin más
- ▶ Elevado de la declaración de variables
- ▶ El ámbito de las variables es la función
- ▶ Si no pones *var* se declara como variable *global*. En *window*

# SVG

- ▶ Imagen vectorial basada en texto
- ▶ **Se puede embeber en el DOM**

```
<svg width="800px" height="600px"> </svg>
```

- ▶ También afectado por CSS
  - ▶ fill
  - ▶ stroke
  - ▶ stroke-width
  - ▶ opacity
- ▶ Mucho material en este manual

