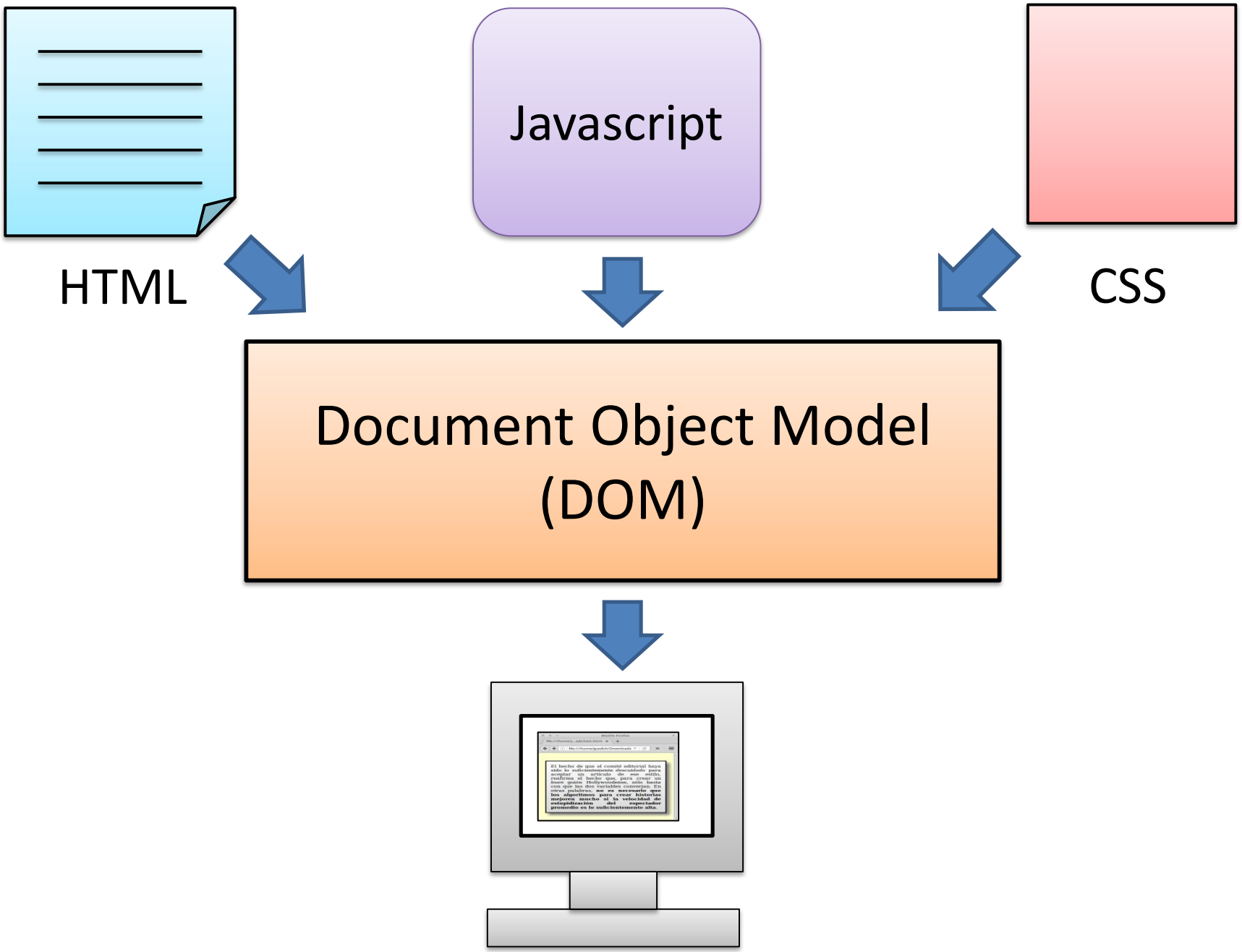
Javascript

Jaime A. Pavlich-Mariscal


Javascript
HTML
CSS
Document Object Model
(DOM)

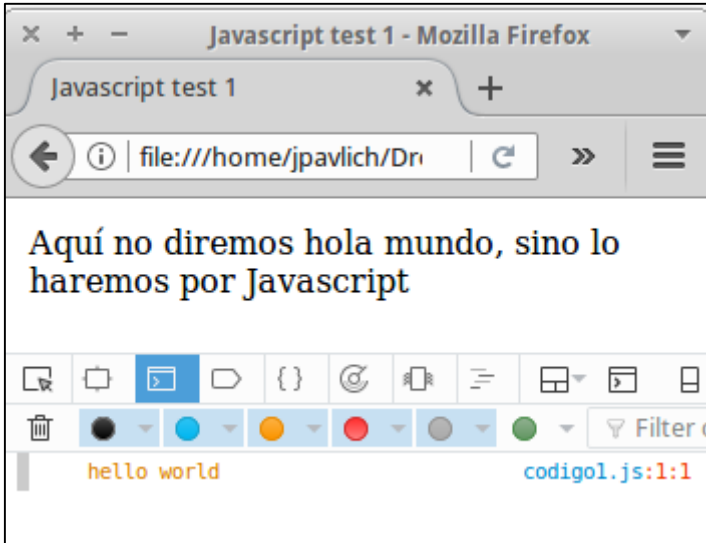
Componentes de una página

- HTML
  - Contenido
  - Qué se va a mostrar
- CSS
  - Presentación
  - Cómo se va a mostrar
- Javascript
  - Comportamiento
  - Cómo cambiará la página a través del tiempo

Ejercicio: Hola Mundo en Javascript

```html
<!DOCTYPE html>
<html>
<head>
    <title>Javascript test 1</title>
    <meta charset="utf-8">
    <link rel="stylesheet" type="text/css" href="formato.css">
    <script type="text/javascript" src="codigo1.js"></script>
</head>
<body>
Aquí no diremos hola mundo, sino lo haremos por Javascript
</body>
</html>
```

```
console.log("hello world");
```



Javascript

- Paradigmas
  - Imperativo
  - Funcional
  - “Orientado a objetos”
- Componentes principales
  - Tipos de datos
  - Sentencias de control
  - Funciones
  - Prototipos

Tipos de datos

Number

- Punto flotante de 64 bits (IEEE 754)
- No existe tipo de dato Integer!

```
console.log(0.3 + 0.6);
```

Strings

- UTF-16
- 中文 español deutsch English हिन्दी العربية português বাংলা русский 日本語 ਪੰਜਾਬੀ 한국어 தமிழ் עברית
- Conversiones
  - `String(2.5) == "2.5"`

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String

Boolean

- Valores: `true`, `false`
- `false`
  - `0`
  - `""`
  - `NaN`
  - `null`
  - `undefined`
- `true`
  - Otros valores

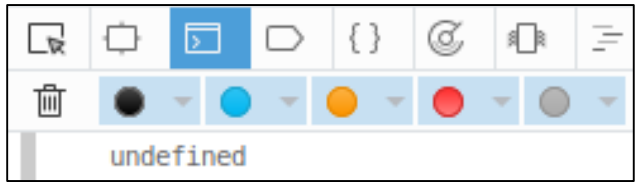
null, undefined y NaN

- `null`: valor nulo

```
var a = null
console.log(a);
```

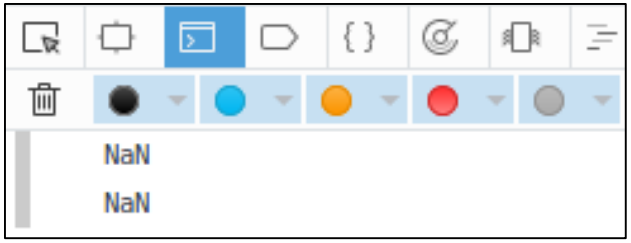
- `undefined`: valor no inicializado

```
var a
console.log(a);
```



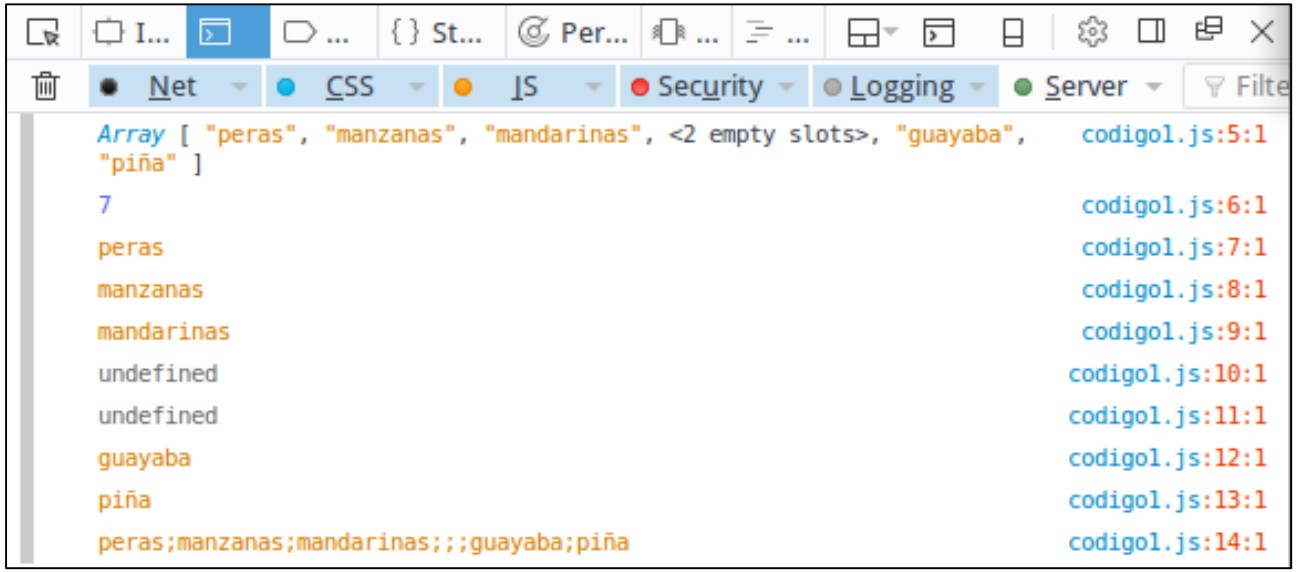
- NaN: “Not a number”

```
console.log(+"hola");
console.log(Math.sqrt(-1));
```



Arrays

```
1  var a = ["peras", "manzanas", "mandarinas"];
2  a[5] = "guayaba";
3  a.push("piña");
4
5  console.log(a);
6  console.log(a.length);
7  console.log(a[0]);
8  console.log(a[1]);
9  console.log(a[2]);
10 console.log(a[3]);
11 console.log(a[4]);
12 console.log(a[5]);
13 console.log(a[6]);
14 console.log(a.join(";"));
```



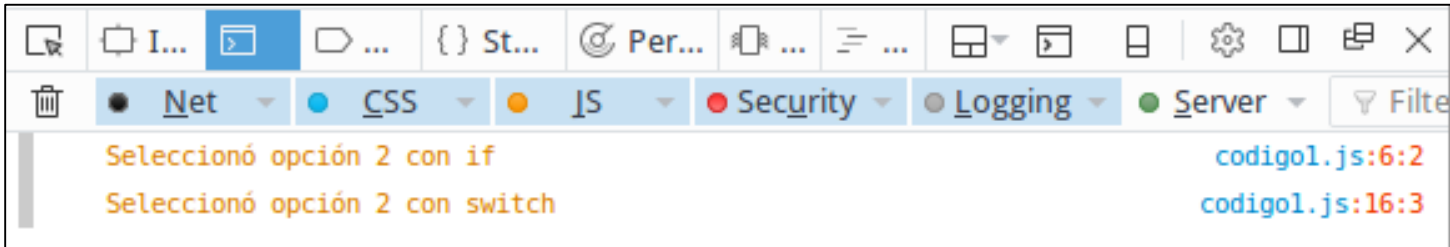
Object

```
var estudiante = {
    nombre: "Pedro",
    cedula: 123456,
    direccion: {
        calle: "7",
        numero: "50-21",
    }
}

console.log(estudiante.nombre);
console.log(estudiante["cedula"]);
console.log(estudiante.direccion.calle);
console.log(estudiante["direccion"]["numero"]);
```

Sentencias de control

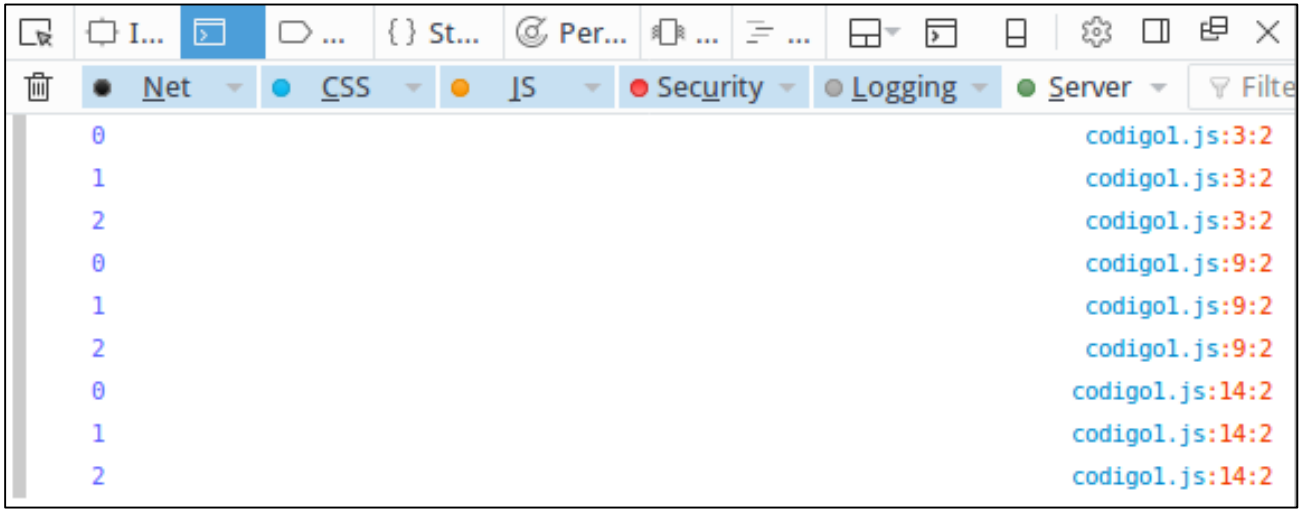
if y switch

```
1  var opcion=2;
2
3  if(opcion==1) {
4      console.log("Seleccionó opción 1 con if");
5  } else if(opcion == 2) {
6      console.log("Seleccionó opción 2 con if");
7  } else if (opcion == 3) {
8      console.log("Seleccionó opción 3 con if");
9  }
10
11 switch(opcion) {
12     case 1:
13         console.log("Seleccionó opción 1 con switch");
14     break;
15     case 2:
16         console.log("Seleccionó opción 2 con switch");
17     break;
18     case 3:
19         console.log("Seleccionó opción 3 con switch");
20     break;
21 }
```



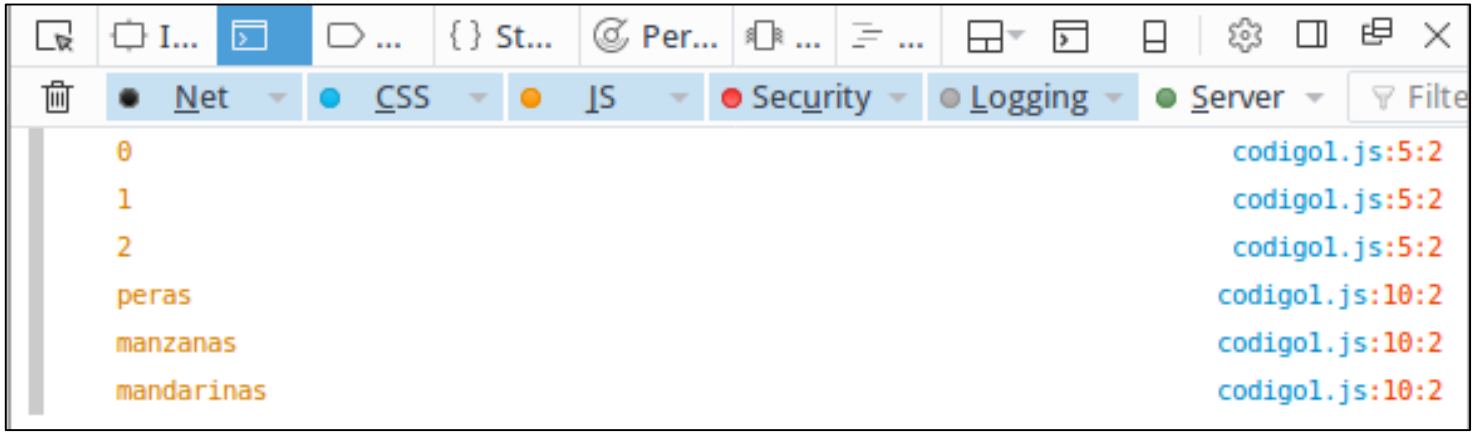
Ciclos

```
1  var i = 0;
2  while(i<3) {
3      console.log(i);
4      i++;
5  }
6
7  var j=0;
8  do {
9      console.log(j);
10     j++;
11 } while (j<3);
12
13 for (var k = 0; k < 3; k++) {
14     console.log(k);
15 }
16
```



Recorrer colecciones

```
 1  var frutas = ["peras", "manzanas", "mandarinas"];
 2
 3  var fruta;
 4  for (fruta in frutas) {
 5      console.log(fruta);
 6  }
 7
 8  var fruta;
 9  for (fruta of frutas) {
10      console.log(fruta);
11  }
```

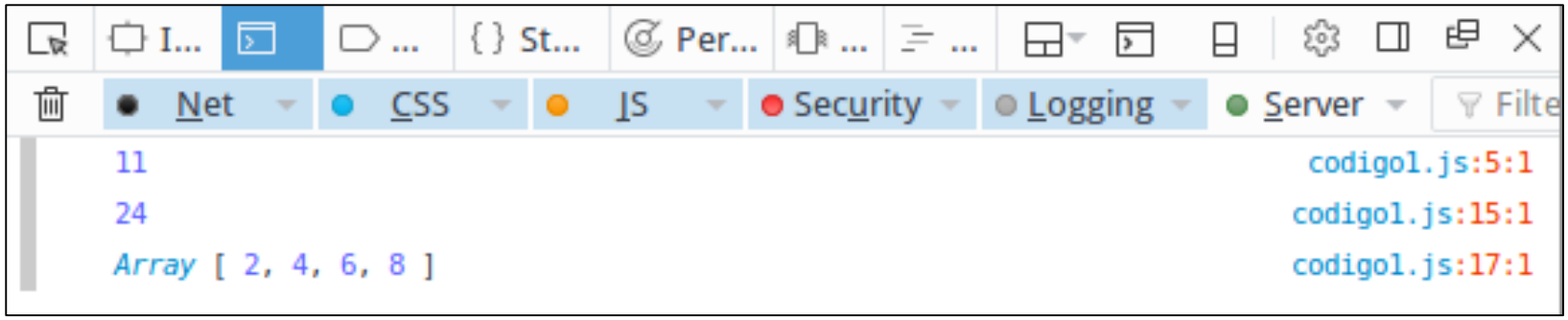


Variables y alcance

```
var x = 10;
function f() {
    var x = 20;
    console.log(x);
}

if(true) {
    var x = 15;
    console.log(x);
}
console.log(x);
f();

let y = 10;
function g() {
    let y = 20;
    console.log(y);
}

if(true) {
    let y = 15;
    console.log(y);
}
console.log(y);
g();
```

Funciones

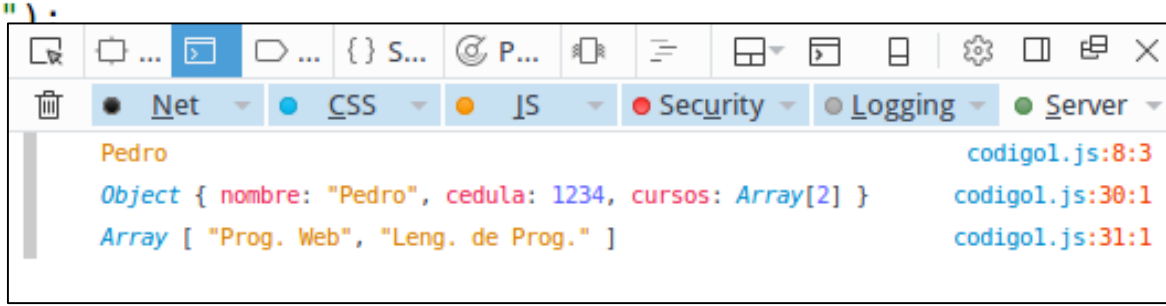
```
1  function use(f, data) {
2      return f(data);
3  }
4
5  console.log(use(function(x) { return x + 1;}, 10));
6
7  function mult(num1,num2) {
8      return num1*num2;
9  }
10
11 function prod(...numbers) {
12     return numbers.reduce(mult, 1);
13 }
14
15 console.log(prod(1,2,3,4));
16
17 console.log([1,2,3,4].map(function(x) {
18     return x*2;
19 }))
```



Prototipos

- Emular Orientación a Objetos

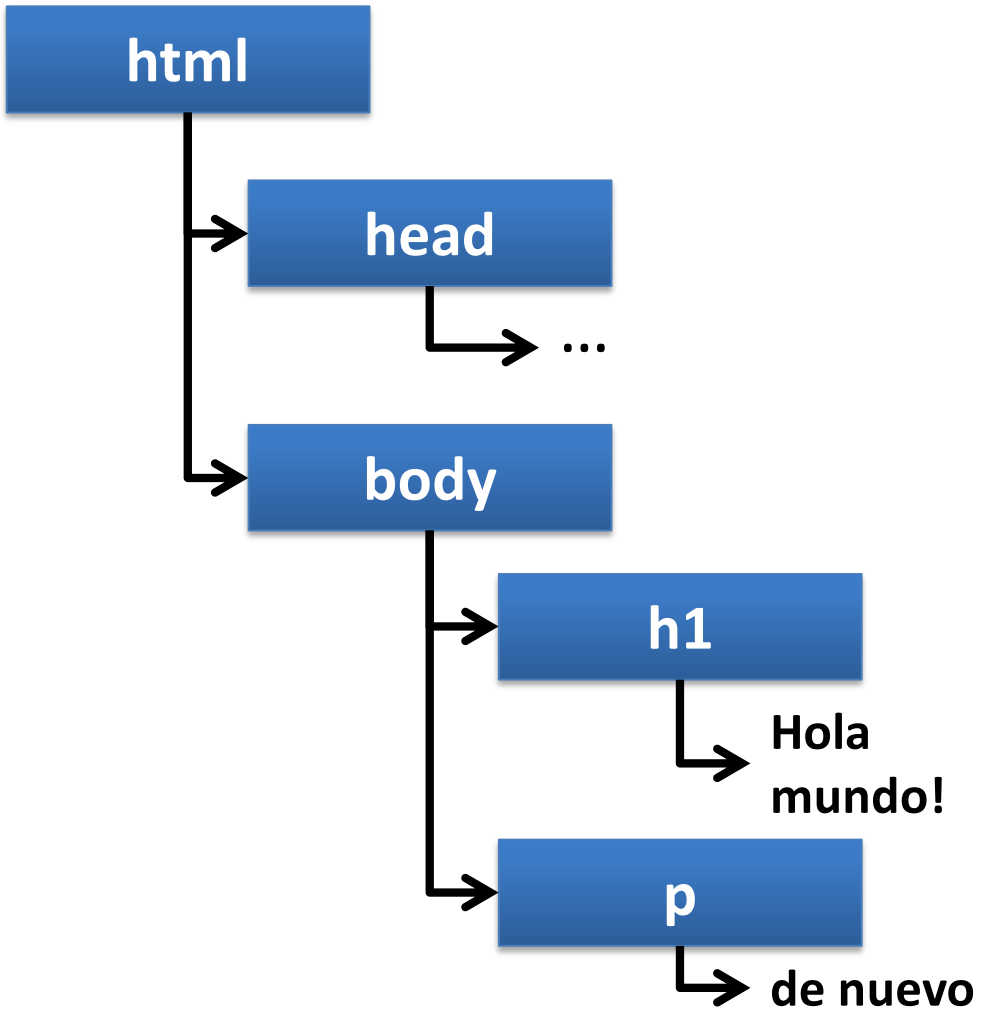
```
1  function Persona(nombre, cedula) {
2      this.nombre = nombre;
3      this.cedula = cedula;
4  }
5
6  Persona.prototype = {
7      printNombre: function() {
8          console.log(this.nombre);
9      }
10 }
11
12 function Estudiante(nombre, cedula) {
13     Persona.call(this, nombre, cedula);
14     this.cursos = [];
15 }
16
17 Estudiante.prototype = Object.create(Persona.prototype);
18
19 Estudiante.prototype.asignarCurso= function(curso) {
20     this.cursos.push(curso);
21 }
22
23 Estudiante.prototype.constructor = Estudiante;
24
25 let e = new Estudiante("Pedro", 1234);
26 e.printNombre();
27 e.asignarCurso("Prog. Web");
28 e.asignarCurso("Leng. de Prog.");
29
30 console.log(e);
31 console.log(e.cursos);
```



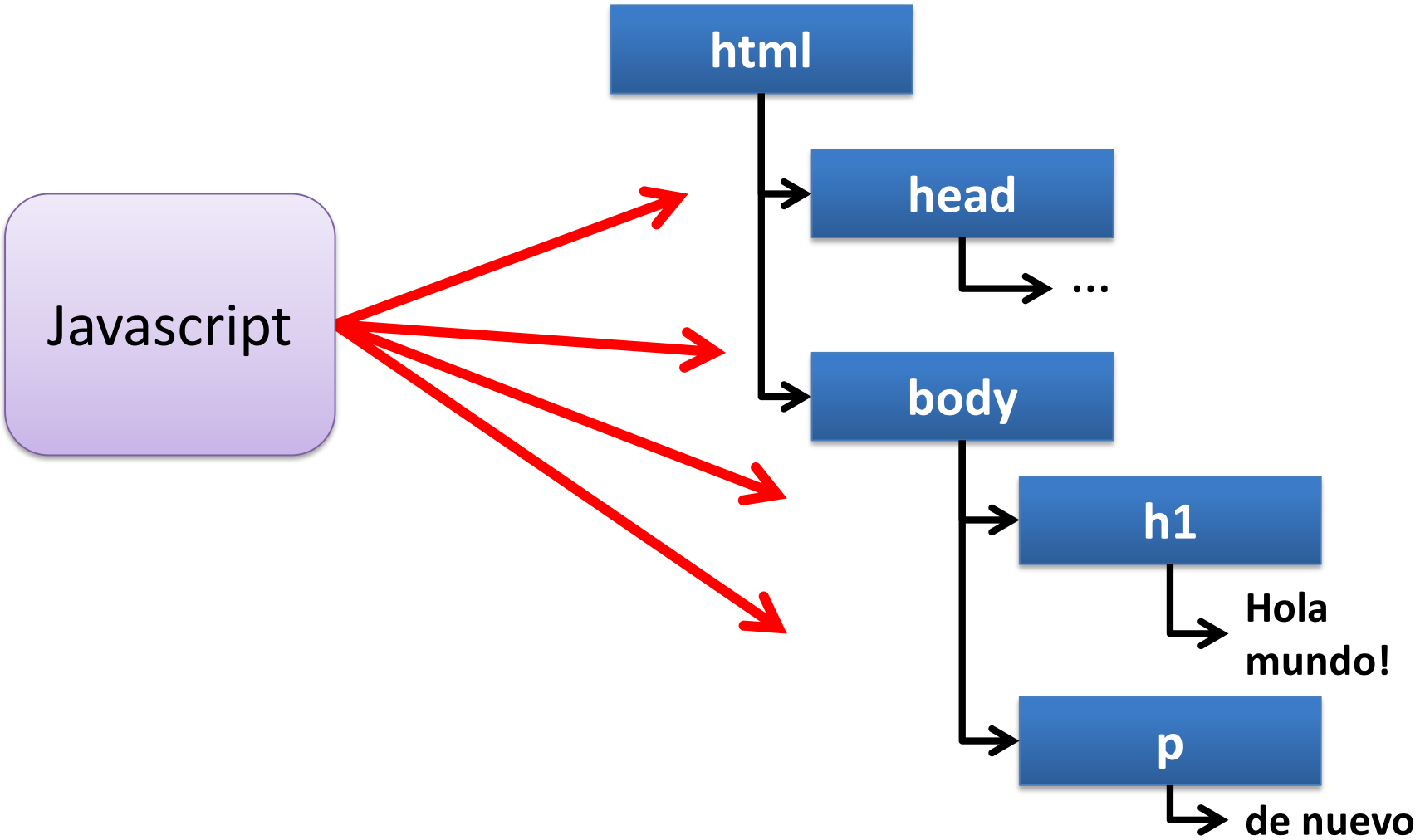
Javascript y el DOM

Document Object Model (DOM)

- Estructura de datos que guarda los contenidos del documento a desplegar
  - Contenido
  - Presentación
- Análogo a un árbol de sintaxis abstracta en lenguajes de programación



Javascript y el DOM



https://developer.mozilla.org/en-US/docs/Web/API/Document

Hola Mundo!

```html
<!DOCTYPE html>
<html>
<head>
    <title>Javascript test 1</title>
    <meta charset="utf-8">
    <link rel="stylesheet" type="text/css" href="formato.css">
    <script type="text/javascript" src="codigo1.js"></script>
</head>
<body>

</body>
</html>
```

```javascript
window.onload = function() {
    var titulo = document.createElement("h1");
    var texto = document.createTextNode("Hola mundo!");
    titulo.appendChild(texto);
    document.body.appendChild(titulo);
}
```

Principales métodos

```
document.getElementById(id)
document.getElementsByTagName(name)
document.createElement(name)
parentNode.appendChild(node)
element.innerHTML
element.style
element.setAttribute
element.getAttribute
element.addEventListener
window.onload
window.scrollTo
```

https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction

Reaccionando a eventos (opción 1)

```
<h1 onclick="mark(this)">Hola Mundo</h1>
<h1 onclick="mark(this)">Hola Mundo</h1>
<h1 onclick="mark(this)">Hola Mundo</h1>
<h1 onclick="mark(this)">Hola Mundo</h1>
<h1 onclick="mark(this)">Hola Mundo</h1>
```

```
function mark(element) {
    element.style.color="red"
}
```

https://developer.mozilla.org/en-US/docs/Web/Events

Reaccionando a eventos (opción 2)

```
<h1>Hola Mundo</h1>
<h1>Hola Mundo</h1>
<h1>Hola Mundo</h1>
<h1>Hola Mundo</h1>
<h1>Hola Mundo</h1>
```

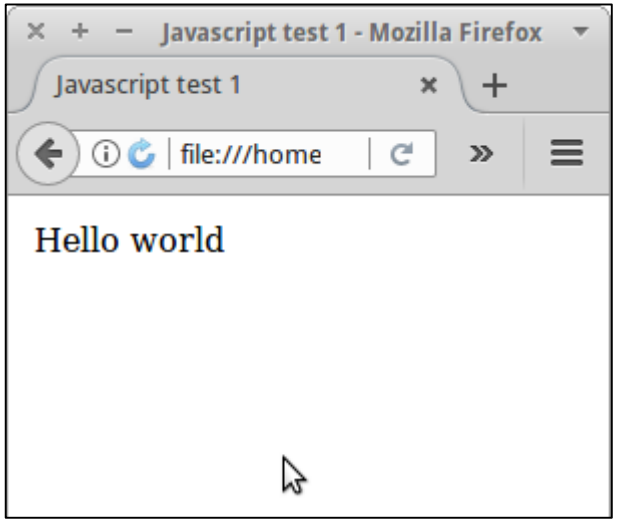
```
function mark(element) {
    element.style.color="red"
}

window.onload = function() {
    var titles = document.getElementsByTagName("h1");
    for (var title of titles) {
        title.addEventListener('click',
            function(event) { mark(event.target); }
        );
    }
};
```

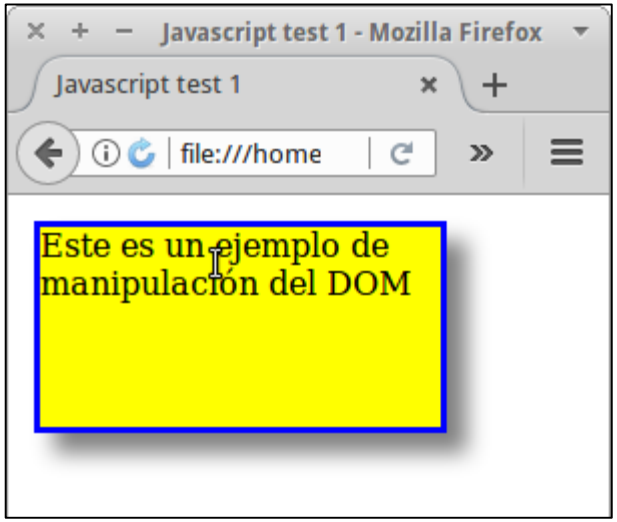
https://developer.mozilla.org/en-US/docs/Web/Events

Manipulando el DOM

```
<div onmouseover="changeThings(this)">
        Hello world</div>
```

```
function changeThings(element) {
    element.style.backgroundColor="yellow";
    element.style.border="solid";
    element.style.borderColor="blue";
    element.style.boxShadow="10px 10px 10px gray";
    element.style.width="200px";
    element.style.height="100px";
    element
        .childNodes[0]
        .nodeValue="Este es un ejemplo de manipulación del DOM";
}
```





Ejercicio: Completar el taller 1

- En toda la página
  - La sección sobre la cual esté posicionado el mouse debe resaltarse cambiando el color de fondo y del borde
- Experiencia laboral, de estudios, premios, producción intelectual
  - Al seleccionar una sección en el índice, el navegador debe deslizarse suavemente hacia dicha sección

Ejercicio: Completar el taller 1

- Crear una “línea de tiempo” donde se puedan escribir mensajes
  - Contenido del mensaje
    - Nombre del Remitente
    - Texto del mensaje
  - Por el momento los mensajes sólo se verán en el cliente (no se almacenan en ningún servidor)
  - El navegador debe validar los datos del mensaje y mostrar mensajes de error al lado del campo mal escrito.
    - Validar que ambos campos no sean vacíos
    - Validar que el texto del mensaje no supere los 300 caracteres