



Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

Client-side JavaScript Parte I

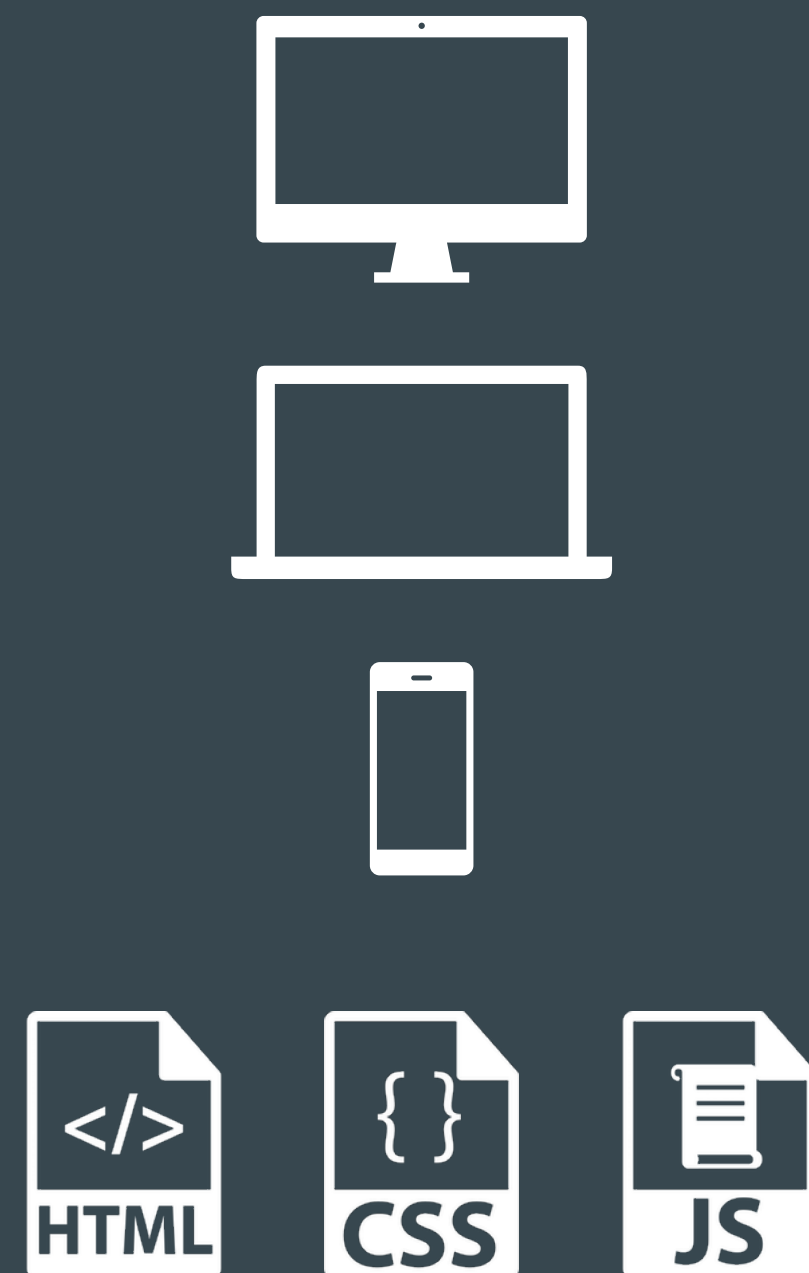
IIC2513 - Tecnologías y Aplicaciones Web

Sebastián Vicencio R.
2do Semestre 2020

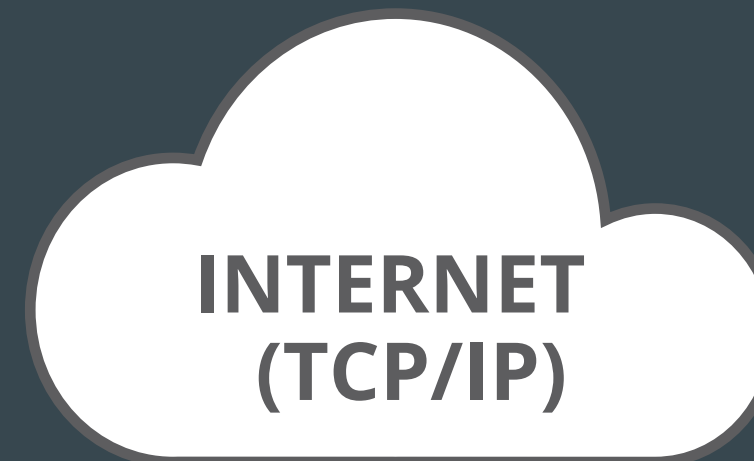
Interacción web

Revisitemos el flujo

CLIENTE



HTTP request



HTTP response

SERVIDOR



JavaScript en el cliente

¿Para qué nos sirve?

Client-side JavaScript

¿Para qué nos sirve?

Manipular HTML y estilos

Interactuar con el usuario



**Generar cambios visuales causados
por interacción con el usuario**

DOM

Document Object Model

DOM

Contexto en el browser

Javascript permite **acceder** a algunas partes del **browser**

- **window**
- **navigator**
- **document**

DOM

¿Recuerdan el árbol HTML?

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<title>My test page</title>
```

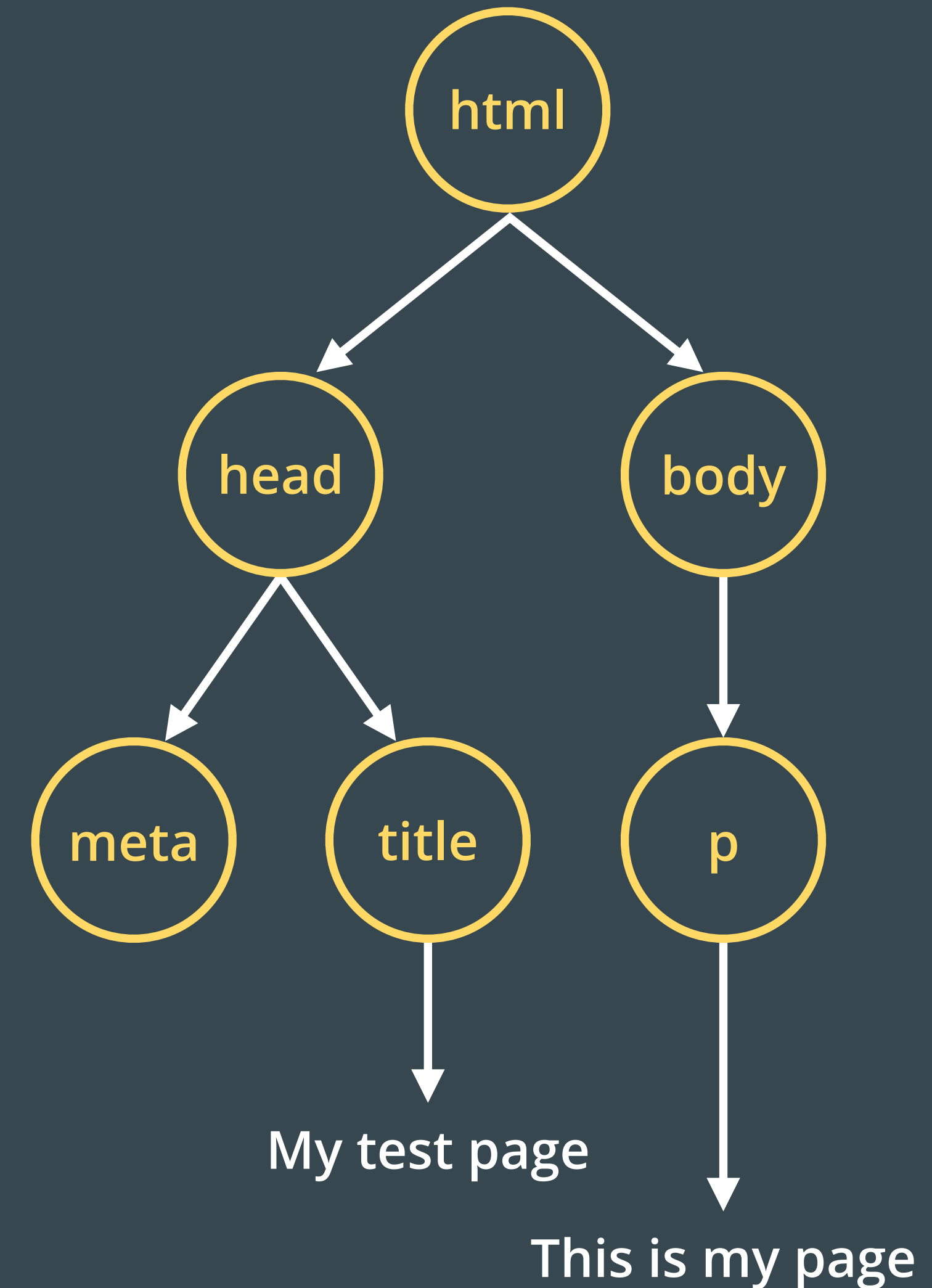
```
</head>
```

```
<body>
```

```
<p>This is my page</p>
```

```
</body>
```

```
</html>
```

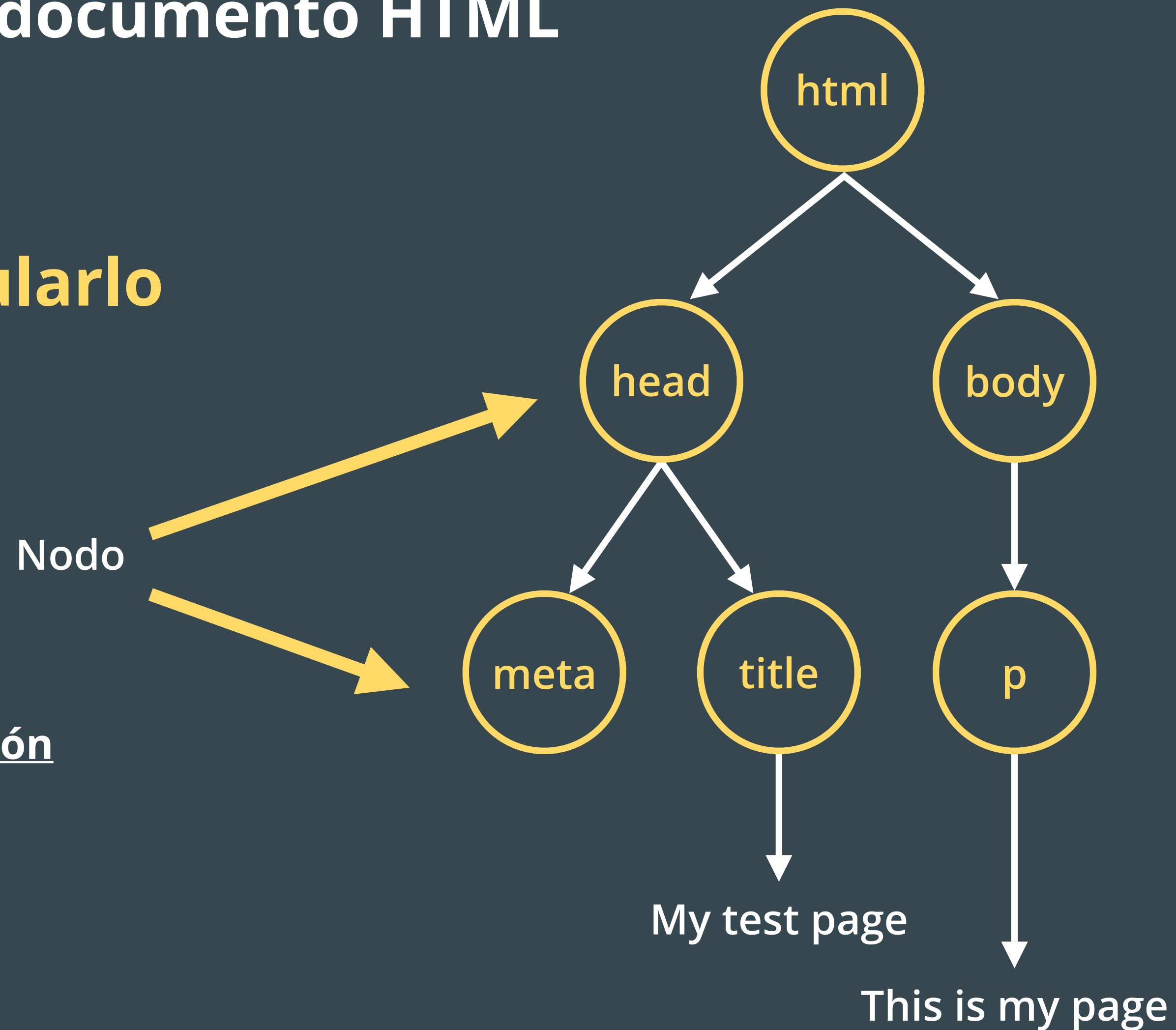


DOM

Representación del documento HTML

Podemos manipularlo
con JS

Ejemplo representación



Manipulación DOM

¿Cómo podemos modificarlo?

Manipulación DOM

Objeto document

El objeto **document** es nuestro **punto de entrada**

- `document.querySelector`
- `document.querySelectorAll`
- `document.getElementById`
- `document.getElementsByTagName`
Name

Manipulación DOM

Modificar elementos

Referencia a elemento para modificarlo

Node.**.**parentNode

Node.**.**childNodes

Node.**.**textContent

element.**.**textContent = 'something'

Manipulación DOM

Crear elementos

Se crea un elemento y luego se inserta

```
const newElement = document.createElement("tagName");  
newElement.textContent = 'I\'m a new element';  
parent.appendChild(newElement);
```

Manipulación DOM

Remover elementos

Se elimina desde el **padre** o desde el **elemento mismo**

```
parent.removeChild(childElement);
```

```
childElement.remove();
```

```
childElement.parentNode.removeChild(childElement);
```

Manipulación DOM

Manipular estilos

Podemos **modificar** propiedad `style` o atributo `class`

```
element.style.backgroundColor = '#FFFFFF';
```

```
element.setAttribute('class', 'some-class');
```

Eventos en el DOM

Interacción con usuario

Eventos en el DOM

¿Qué son los eventos?

Sucesos que ocurren en una **página web** y/o en la **ventana del browser**

- Click en un botón
- Hover en un elemento
- Presionar una tecla
- Submit de un formulario

Eventos en el DOM

Event handlers

Es posible **responder a eventos** con una **función**

```
handlerFunction() {  
    // Respond to an event  
}
```

Formas de responder a eventos

Event handler properties

Asociamos la función a **propiedad del evento**

```
element.onclick = handlerFunction() {  
    // Respond to click event  
}
```

Formas de responder a eventos

Inline

Asociamos función a atributo HTML del elemento
¡Evitar!

```
<button onclick="handlerFunction()"></button>
```

No mezclar HTML con JS
Ineficiente
Poco mantenible

Formas de responder a eventos

addEventListener

Asociamos función mediante el **método addEventListener**

```
element.addEventListener('event', handlerFunction);
```

```
element.removeEventListener('event', handlerFunction);
```

Eventos

Otros conceptos

Event objects

```
element.addEventListener('event', function(e) {  
    console.log(e.target);  
});
```

Prevenir comportamiento predeterminado

```
element.addEventListener('event', function(e) {  
    e.preventDefault();  
});
```

Incluir JavaScript

¿Cómo agregarlo al HTML?

Cómo incluir JavaScript

Etiqueta script

Internal

```
<script>  
    // JS Code  
</script>
```

External

```
<script src="path/to/script"></script>
```

Cómo incluir JavaScript

Evento DOMContentLoaded

Necesitamos que DOM se cargue
antes de interactuar con él

```
document.addEventListener("DOMContentLoaded", function() {  
    // Code ready to handle DOM  
});
```


Ejemplos

Likes

Lista

Referencias

- MDN - "Manipulating documents"
- MDN - "Document Object Model (DOM)"
- MDN - "Introduction to events"
- MDN - "Event reference"
- Eloquent JavaScript: "The Document Object Model"
- Eloquent JavaScript: "Handling Events"