




Javascript Avanzado I

 2024-03   35 min.

Optional Chaining

Optional Chaining

```
function saludos(user) {  
  console.log("Hola " + user.name);  
}  
  
let user = {  
  name: "El chancclas",  
};  
  
saludos(user); // ??
```

Optional Chaining

```
function saludos(user) {  
  console.log("Hola " + user.name);  
}  
  
let user = {  
  name: "El chanclas",  
};  
  
saludos(user); // "Hola El chanclas"
```

Optional Chaining

```
function saludos(user) {  
  console.log("Hola " + user.name);  
}  
  
let user = {  
  // empty  
};  
  
saludos(user); // ??
```

Optional Chaining

```
function saludos(user) {  
  console.log("Hola " + user.name);  
}  
  
let user = {  
  // empty  
};  
  
saludos(user); // "Hola undefined"
```

Optional Chaining

```
function saludos(user) {  
  console.log("Hola " + user.name);  
}  
  
//  
// no user  
//  
  
saludos(); // ??
```

Optional Chaining

```
function saludos(user) {  
  console.log("Hola " + user.name);  
}  
  
//  
// no user  
//  
  
saludos(); // ⚠
```


Optional Chaining

```
function saludos(user) {  
  console.log("Hola " + user?.name);  
}  
  
//  
// no user  
//  
  
saludos(); // "Hola undefined"
```

Optional Chaining

```
if (user !== undefined) {  
  return user.name  
} else {  
  return undefined  
}
```

```
return user?.name
```

Parametros Predeterminados

Parametros Predeterminados

```
function saludos(name) {  
  console.log("Hola " + name);  
}  
  
let user = {  
  name: "El chancas",  
};  
  
saludos(user.name); // "Hola El chancas"
```

Parametros Predeterminados

```
function saludos(name) {  
  console.log("Hola " + name);  
}  
  
let user = {  
  // aquí no hay nada  
};  
  
saludos(user.name); // "Hola undefined"
```

Parametros Predeterminados

```
function saludos(name) {  
  console.log("Hola " + name);  
}  
  
//  
// ahora declarado como indefinido  
let user = undefined;  
  
saludos(user.name); // ⚠
```

Parametros Predeterminados

```
function saludos(name) {  
  console.log("Hola " + name);  
}  
  
//  
// ahora declarado como indefinido  
let user = undefined;  
  
saludos(user?.name); // "Hola undefined"
```

Parametros Predeterminados

```
function saludos(name = "usuario") {  
  console.log("Hola " + name);  
}  
  
//  
// sigue siendo indefinido  
let user = undefined;  
  
saludos(user?.name); // "Hola usuario"
```


Parametros Predeterminados

```
function saludos(name = "usuario") {  
  console.log("Hola " + name);  
}  
  
let user = {  
  // nadita  
};  
  
saludos(user.name); // "Hola usuario"
```

Parametros Predeterminados

```
function suma(a = 0, b = 0) {  
  return a + b;  
}  
  
console.log(suma(1, 2));  
console.log(suma(1));  
console.log(suma(undefined, 2));  
console.log(suma());
```

Parametros Predeterminados

```
function suma(a = 0, b = 0) {  
  return a + b;  
}  
  
console.log(suma(1, 2));  
console.log(suma(1));  
console.log(suma(undefined, 2));  
console.log(suma());
```

Parametros Predeterminados

```
function suma(a = 0, b = 0) {  
  return a + b;  
}  
  
console.log(suma(1, 2)); // 3  
console.log(suma(1));  
console.log(suma(undefined, 2));  
console.log(suma());
```

Parametros Predeterminados

```
function suma(a = 0, b = 0) {  
  return a + b;  
}  
  
console.log(suma(1, 2)); // 3  
console.log(suma(1)); // 1  
console.log(suma(undefined, 2));  
console.log(suma());
```

Parametros Predeterminados

```
function suma(a = 0, b = 0) {  
  return a + b;  
}  
  
console.log(suma(1, 2)); // 3  
console.log(suma(1)); // 1  
console.log(suma(undefined, 2)); // 2  
console.log(suma());
```

Parametros Predeterminados

```
function suma(a = 0, b = 0) {  
  return a + b;  
}  
  
console.log(suma(1, 2)); // 3  
console.log(suma(1)); // 1  
console.log(suma(undefined, 2)); // 2  
console.log(suma()); // 0
```

Parametros Predeterminados

```
function suma(a = 15, b = 18) {  
  return a + b;  
}  
  
console.log(suma(4, true));  
console.log(suma(3, false));
```


Parametros Predeterminados

```
function suma(a = 15, b = 18) {  
  return a + b;  
}  
  
console.log(suma(4, true)); // 5  
console.log(suma(3, false)); // 3
```

Null Coalescing Operator

Null Coalescing Operator

```
function saludo(nombre1, nombre2) {  
  if (nombre2) {  
    return "Hola " + nombre2;  
  } else if (nombre1) {  
    return "Hola " + nombre1;  
  }  
  
  return "Hola abominacion";  
}  
  
console.log(saludo("Experimento 626", "Stitch")); // ??  
console.log(saludo("Experimento 626")); // ??  
console.log(saludo()); // ??
```

Null Coalescing Operator

```
function saludo(nombre1, nombre2) {  
  return "Hola " + (nombre2 ?? nombre1 ?? "abominacion");  
}
```

```
console.log(saludo("Experimento 626", "Stitch")); // ??  
console.log(saludo("Experimento 626")); // ??  
console.log(saludo()); // ??
```

Null Coalescing Operator

```
function saludo(nombre1, nombre2) {  
  return "Hola " + (nombre2 ?? nombre1 ?? "abominacion");  
}
```

```
console.log(saludo("Experimento 626", "Stitch")); // ??  
console.log(saludo("Experimento 626", "undefined")); // ??  
console.log(saludo(null, undefined)); // ??
```

Template Strings

Template Strings

```
function saludo(nombre1, nombre2) {  
  return "Hola " + (nombre2 ?? nombre1 ?? "abominacion");  
}  
  
console.log(saludo("Experimento 626", "Stitch")); // ??  
console.log(saludo("Experimento 626")); // ??  
console.log(saludo()); // ??
```

Template Strings

```
function saludo(nombre1, nombre2) {  
  return `Hola ${nombre2 ?? nombre1 ?? "abominacion"}`;  
}  
  
console.log(saludo("Experimento 626", "Stitch")); // ??  
console.log(saludo("Experimento 626")); // ??  
console.log(saludo()); // ??
```


Template Strings

`

"Back tick"

Template Strings

Alt Gr + }

Template Strings

```
function saludo(nombre1, nombre2) {  
  return `Hola ${nombre2 ?? nombre1 ?? "abominacion"}`;  
}  
  
console.log(saludo("Experimento 626", "Stitch")); // ??  
console.log(saludo("Experimento 626")); // ??  
console.log(saludo()); // ??
```

Intermisión

Manipulación de Strings

Nota

Los metodos de manipulación de string **retornan** la string modificada.

.split()

```
let text = "Apple, Banana, Kiwi";  
  
console.log(text.split(",")); // ??
```

.split()

```
let text = "Apple, Banana, Kiwi";  
  
console.log(text.split(",")); // ["Apple", " Banana", " Kiwi"]
```


.split()

```
let text = "Apple, Banana, Kiwi";  
  
console.log(text.split(", ")); // ["Apple", "Banana", "Kiwi"]
```

.split()

```
function separarApellidos(apellidos) {  
  const apellidos = [];  
  let apellidoActual = "";  
  
  for (let i = 0; i < apellidos.length; i++) {  
    if (apellidos[i] === " ") {  
      apellidos.push(apellidoActual);  
      apellidoActual = "";  
    } else {  
      apellidoActual += apellidos[i];  
    }  
  }  
  
  return apellidos;  
}
```

```
function separarApellidos(apellidos) {  
  return apellidos.split(" ");  
}
```

.substring()

```
let text = "Apple, Banana, Kiwi";

// el segundo parametro es la posicion de fin
console.log(text.substring(7)); // "Banana, Kiwi"
console.log(text.substring(7, 13)); // "Banana"
console.log(text.substring(7).substring(8)); // "Kiwi"
```

.substr()

```
let text = "Apple, Banana, Kiwi";

// el segundo parametro es la cantidad de caracteres
console.log(text.substr(7)); // "Banana, Kiwi"
console.log(text.substr(7, 6)); // "Banana"
console.log(text.substr(7).substr(0, 6)); // "Banana"
```

.slice()

```
let text = "Apple, Banana, Kiwi";

// el segundo parametro es la posicion de fin
console.log(text.slice(7)); // "Banana, Kiwi"
console.log(text.slice(7, 13)); // "Banana"
console.log(text.slice(7).slice(0, 6)); // "Banana"
```

.slice()

```
let text = "Apple, Banana, Kiwi";

// el segundo parametro es la posicion de fin
console.log(text.slice(-12)); // "Banana, Kiwi"
console.log(text.slice(-12, 13)); // "Banana"
console.log(text.slice(-12, -6)); // "Banana"
console.log(text.slice(7, -6)); // "Banana"
console.log(text.slice(-12).slice(-4)); // "Kiwi"
```

.trim()

```
let text = "  Apple, Banana, Kiwi  ";  
  
console.log(text.slice(7, 13)); // ", Bana"
```

.trim()

```
let text = "  Apple, Banana, Kiwi  ";  
text = text.trim();  
  
console.log(text.slice(7, 13)); // "Banana"
```


.trim()

```
let text = "  Apple, Banana, Kiwi  ";  
  
console.log(text.trim().slice(7, 13)); // "Banana"
```

.trimStart(), .trimEnd()

```
let text = "  Apple, Banana, Kiwi  ";  
  
console.log(text.trimStart().trimEnd().slice(7, 13)); // "Banana"
```

.padStart(), .padEnd()

```
let text = "Kiwi";  
  
console.log(text.padStart(" ").padEnd(" ")); // "  Kiwi  "
```

Object.keys()

Object.keys()

```
const obj = {  
  key: "value",  
};  
  
console.log(Object.keys(obj)); // ["key"]
```

Object.keys()

```
const user = {  
  firstname: "Marlene",  
  lastname: "Duarte",  
};  
  
console.log(Object.keys(user)); // ["firstname", "lastname"]
```

.flat()

```
const numbers = [[4, 5], [2, 1], [1], [7]];

console.log(numbers.flat()); // [4, 5, 2, 1, 7]
```

.join()

```
const numbers = [4, 5, 2, 1, 7];  
  
console.log(numbers.join(", ")); // "4, 5, 2, 1, 1, 7"
```


.includes()

```
const numbers = [4, 5, 2, 1, 7];  
  
console.log(numbers.includes(3)); // false  
console.log(numbers.includes(7)); // true
```



Inserte Disco 2