Undefined Academy

Bootcamp de JavaScript Full-stack

Semana 2, Clase 2:

JavaScript 101

Metodología de trabajo

Formato de las sesiones ⏱️ (1h 30m)

- `10 min` Revisión de ejercicio anterior
- `60 min` Presentación teórica-práctica
- `10 min` Preguntas y discusión

Herramientas del Bootcamp 🔧

- **Twitch**: https://undf.sh/en-vivo
- **Discord**: https://undf.sh/discord
- **Notion**: https://undf.sh/base

Agenda

- Objetos y sus propiedades en JavaScript
- Funciones puras y mutación en JavaScript
- Los operadores lógicos en JavaScript
- Condicionales y ciclos en JavaScript
- El alcance en JavaScript

Revisión de ejercicio anterior

Objetos y sus propiedades en JavaScript

Objetos y sus propiedades en JavaScript

```
const human = {
    name: "Guillermo",
    lastname: "Rodas",
    age: 32
}

console.log(human);
```

Podemos agregar nuevas propiedades

```
human.id = Symbol("glrodasz");
human.blonde = true;

console.log(human);
```

Se puede acceder a las propiedades con el operador "punto" o "corchete".

```
const designer = {
    name: "Juan",
    lastname: "Garces",
    "The best Designer": true
}

console.log(designer["name"])
```

Las propiedades se pueden cambiar en el tiempo

```
human.blonde = false;
human.age = human.age - 10;
```

Podemos eliminar las propiedades con el operador "delete"

```
delete human.lastname
delete human.age

console.log(human)
```

Funciones puras y mutación en JavaScript

Funciones puras y mutación en JavaScript

Function Statement[1]

```
function walk() {
    console.log("I'm walking");
}

walk()
```

1. Sentencia de función/ Declaración

Funciones puras y mutación en JavaScript

Function Expression[1]

```
const walk = function () {
    console.log("I'm walking");
}

walk()
```

1. Expresión de función

Funciones puras y mutación en JavaScript

Funciones Puras

1. **No tiene efectos secundarios**: No modifica ningún estado fuera de su alcance, como variables globales o referencias.
2. **Es determinista**: Esto significa que dada la misma entrada, siempre producirá la misma salida.

Funciones Puras

Ejemplo:

```
function sum(x, y) {
    return x + y;
}

sum(2, 1)
```

`x` y `y` son los **Parámetros** de la función, mientras que en la ejecución `2` y `1` pasan a ser los **Argumentos**.

Copia por Valor y copia por Referencia

```
let name = "Guillermo";
let nickname = nombre;
name = "William";

console.log(name)
console.log(nickname)
```

Los primitivos siempre se copian por valor

Copia por Valor y copia por Referencia

```
let human = { name: "Guillermo" };
let alien = human;
alien.name = "Omrelliug";

console.log(human)
console.log(alien)
```

Los objetos siempre se copian por referencia

¿Es una función pura o impura?

```
function changeName(person, newName) {
    person.name = newName;
    return person;
}

changeName(human, "Wilhelm")
```

 ¿Cómo hacer una copia de un objeto de manera pura?

Los operadores lógicos en JavaScript

Los operadores lógicos en JavaScript

- OR: `a || b || c` retorna el primer valor verdadero
- AND: `a && b && c` retorna el primer valor falso
- NOT: `!a` retorna el valor contrario booleano

Condicionales y ciclos en JavaScript

Condicionales y ciclos en JavaScript

If Statement

```
if (age >= 18) {
    console.log("Adult")
} else if (age >= 10) {
    console.log("Young")
} else {
    console.log("Baby")
}
```

Condicionales y ciclos en JavaScript

Operador Ternario ?

```
let msg = age >= 18 ? "Adult" : "Young";
```

Condicionales y ciclos en JavaScript

¿Cómo haríamos una lista de frutas?

```
const fruits = {
    0: "Banana",
    1: "Orange",
    2: "Apple"
}

console.log(fruits[0])
```

Condicionales y ciclos en JavaScript

Los Arreglos[1]

```
const fruits = [
    "Banana",
    "Orange",
    "Apple"
]

console.log(fruits[0])
typeof fruits
```

1. Arrays

Condicionales y ciclos en JavaScript

El ciclo/bucle `for`

```
for (let i = 0; i < 3; i++) {
  console.log(i);
}
```

El alcance en JavaScript

```
{
  let msg = "Hello world";
  console.log(msg);
}

let msg = "Olá mundo"
let planet = "mars"

function scope() {
    let msg = "Hello world";
    let planet = "earth"
    console.log(msg)

    if (msg) {
        let msg = "Hola mundo";
        console.log(msg, planet)
    }

    msg = "Hej världen"
    console.log(msg)
}

scope()
console.log(msg, planet)
```

Recursos 📚

- https://es.javascript.info/
- https://developer.mozilla.org/es/docs/Learn/JavaScript
- https://carlosazaustre.es/libros/aprendiendo-javascript
- https://exploringjs.com/impatient-js/
- https://www.freecodecamp.org/espanol/news/aprende-javascript-curso-completo-desde-cero/

Calentamiento

Escribir un objeto "humano" que te represente.

Ejemplo:

```
const human = {
    name: "Guillermo",
    blonde: true
};
```

Ejercicio

¿Cómo puedo implementar una expresión para verificar si un valor es un array?

Ejemplo: `typeof arr === "object"`

¿Preguntas?