¿Qué diferencia a Javascript de cualquier otro lenguaje de programación?

Javascript es el único lenguaje legible por navegadores directamente. Otros lenguajes como Python se alojan en un servidor y requieren ser interpretados por éste antes de ejecutarse, cosa que en Javascript no es necesaria porque podemos ejecutarlo, como ya he dicho, directamente desde el navegador, sin necesidad de instalar nada.

¿Cuáles son algunos tipos de datos JS?

-Booleano: Devuelven un valor True (verdadero) o False (falso). Por ejemplo, un condicional 10 > 9 devolvería True, mientras que si fuera 9 > 10 devolvería False.

-String: Son cadenas de texto. Van entre comillas dobles o simples (“”, ‘’), aunque al escribir en inglés cosas como *’s* deben utilizarse las dobles para evitar que la cadena de texto se rompa donde no debe.

-Número: Cualquier tipo de número, ya sea positivo o negativo. Se escriben tal cual (42, -10, 357...) y es posible efectuar operaciones con ellos, como sumas (+), restas (-), multiplicaciones (*) y divisiones (/). Además también se pueden hacer otras operaciones como por ejemplo usar los módulos (%), que dan el resto de una división. Esto es muy útil para saber si un número es par o no, ya que si es par el resto siempre será cero.

-Objeto: Son estructuras de datos que contienen múltiples valores. Se escriben entre { }. Un ejemplo podría ser:

```
let persona = {
    nombre: “Juan”,
    edad: 42,
    idioma: “español” }
```

Si quiero acceder por ejemplo al idioma escribiría algo como:

```
console.log(persona.idioma)
```

¿Cuáles son las tres funciones de String en JS?

Hay muchas operaciones que se pueden hacer con cadenas de texto. Algunas de ellas son:

-**Length**: Permite saber cuál es la longitud de una cadena de texto. Imaginemos que tenemos la siguiente cadena de texto:

```
let string = “Soy una cadena de texto”
```

Si le pasamos la función length así

```
console.log(string.length)
```

nos devuelve 13, que es su longitud.

-**Concat**: Permite concatenar (unir) dos o más cadenas. Si además del string anterior tuviéramos algo como

```
let StringDos = “ y yo soy otra cadena de texto”
```

podríamos hacer lo siguiente:

```
console.log(string.concat(StringDos))
```

esto da como resultado

“Soy una cadena de texto y yo soy otra cadena de texto”.

Este proceso se puede simplificar aún más haciendo simplemente

```
string + StringDos
```

y tendríamos el mismo resultado.

-**Replace**: Sustituye una palabra por otra dentro de la cadena. El primer argumento es el valor a sustituir, y el segundo la palabra que se inserta:

```
string.replace(“texto”, “sonido”)
```

Esto da como resultado “Soy una cadena de sonido”.

¿Qué es un condicional?

Un condicional se ejecuta si se cumple una condición. También puede indicarse que se realice otra acción si la condición no se cumple. Por ejemplo, si vemos por la calle el semáforo en rojo no cruzaremos, pero sí lo haremos si está en verde. Traducido a código esto queda así:

```
let semaforo = "rojo"
        if (semaforo == "rojo") {
                console.log("No cruzar"); } else {
                 console.log("Puede cruzar");
        }
}
```

Es posible anidar condicionales para crear, por ejemplo, programas que tengan en cuenta condiciones más complejas, pero esto volvería el código más difícil de leer y es preferible, si hay condiciones complejas, usar un switch, cuya estructura (volviendo al ejemplo del semáforo) es como sigue:

```
        switch (semaforo) {
                case rojo:
                "No puedes cruzar"
        break;
                case verde:
                "Puedes cruzar"
      break;
  default: amarillo
                “Ve con cuidado”
    break; }
```

Es importante notar que la sentencia break es importantísima, porque si no el programa no sabe cuándo parar, y así, tras un break, podemos poner todas las condiciones que queramos sin que el código sea difícil de leer. Por último, es posible colocar un valor por defecto (default), que estará disponible si no se indica nada como parámetro.

¿Qué es un operador ternario?

Un operador ternario es una forma de escribir condicionales en una sola línea de código. Volvamos al ejemplo del semáforo. Usando un operador ternario quedaría así:

```
semaforo == “rojo” ? “No puedes cruzar” : “Puedes cruzar”
```

La primera parte antes del signo de interrogación es la condición que se evalúa, y tras él están los resultados dependiendo de qué variable tengamos, siendo el de la izquierda el resultado de if y el de la derecha el de else.

¿Cuál es la diferencia entre una declaración de función y una expresión de función?

Una función declarada es una función “normal”, como por ejemplo:

```
function saludar() {
  return ("Hola, mundo!");
}
```

Y una función expresada es una función que se guarda dentro de una variable:

```
const saludo = function saludar() {
  return ("Hola, mundo!");
}
```

¿Qué es la palabra clave "this" en JS?

La palabra reservada this hace referencia al objeto actual con el que se está trabajando. Dependiendo de dónde se invoque puede referirse a diferentes objetos:

- En métodos de objetos se refiere al objeto
- Por sí solo, se refiere al scope global
- En una función también se refiere al scope global, excepto en el modo estricto, donde será indefinida
- En eventos, this se refiere al elemento HTML que recibe el evento