

4.

OPERACIONES BÁSICAS



OPERADORES ARITMÉTICOS

Operador	Nombre	Propósito	Ejemplo
+	Adición	Suma dos números juntos.	$6 + 9$
-	Resta	Resta el número de la derecha del de la izquierda.	$20 - 15$
*	Multiplicación	Multiplica dos números juntos.	$3 * 7$
/	División	Divide el número de la izquierda por el de la derecha.	$10 / 5$
%	Sobranter (también llamado módulo)	Retorna el restante después de dividir el número de la izquierda en porciones enteras del de la derecha.	$8 \% 3$ (retorna 2, como tres está dos veces en 8, quedando 2 restantes.)

OPERADORES INCREMENTO Y DECREMENTO

- incremento:

```
let incremento = 3;
```

```
incremento++;
```

- decremento:

```
let decremento = 5;
```

```
decremento—;
```

OPERADORES DE ASIGNACIÓN

Operador	Nombre	Propósito	Ejemplo	Atajo de
<code>+=</code>	Adición asignación	Suma el valor de la derecha al valor de la variable de la izquierda y retorna el nuevo valor	<code>x = 3;</code> <code>x += 4;</code>	<code>x = 3;</code> <code>x = x + 4;</code>
<code>-=</code>	Resta asignación	Resta el valor de la derecha, del valor de la variable de la izquierda y retorna el nuevo valor.	<code>x = 6;</code> <code>x -= 3;</code>	<code>x = 6;</code> <code>x = x - 3;</code>
<code>*=</code>	Multiplicación asignación	Multiplica el valor de la variable en la izquierda por el valor en la derecha y retorna el nuevo valor.	<code>x = 2;</code> <code>x *= 3;</code>	<code>x = 2;</code> <code>x = x * 3;</code>
<code>/=</code>	División asignación	Divide el valor de la variable en la izquierda por el valor de la derecha y retorna el nuevo valor.	<code>x = 10;</code> <code>x /= 5;</code>	<code>x = 10;</code> <code>x = x / 5;</code>

OPERADORES DE COMPARACIÓN

Operador	Nombre	Propósito	Ejemplo
<code>===</code>	Igual estricto	Comprueba si los valores izquierdo y derecho son idénticos entre sí	<code>5 === 2 + 4</code>
<code>!==</code>	Igual no-estricto	Comprueba si los valores izquierdo y derecho no son idénticos entre sí	<code>5 !== 2 + 3</code>
<code><</code>	Menor que	Comprueba si el valor izquierdo es menor que el derecho.	<code>10 < 6</code>
<code>></code>	Mayor que	Comprueba si el valor izquierdo es mayor que el derecho.	<code>10 > 20</code>
<code><=</code>	Menor o igual a	Comprueba si el valor izquierdo es menor o igual que el derecho.	<code>3 <= 2</code>
<code>>=</code>	Mayor o igual a	Comprueba si el valor izquierdo es mayor o igual que el derecho..	<code>5 >= 4</code>



MANEJO DE STRINGS

- `toLowerCase` \Rightarrow `'Hola'.toLowerCase();`
- `toUpperCase` \Rightarrow `'Hola'.toUpperCase();`
- `replace` \Rightarrow `'Hola Mundo'.replace('Mundo', 'Amigo');`
- `split` \Rightarrow `'Hola Mundo'.split(' ');`

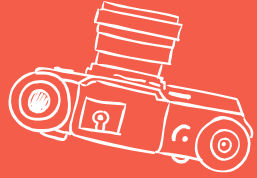
MANEJO DE STRINGS

- `startsWith` `=> 'Hola Mundo'.startsWith('Hola');`
- `endsWith` `=> 'Hola Mundo'.endsWith('Mundo');`
- `trim` `=> ' Hola Mundo '.trim();`
- `charAt` `=> 'Hola'.charAt(2);`

STRINGS Y NUMBERS

1. `"hello" + 123;`
2. `"19" + "95";`
3. `Number("1234");`
4. `let myNumber = 123; myNumber.toString();`
5. `parseInt("50"); parseInt("50.55");`
6. `parseFloat("50"); parseFloat("50.55");`
7. `String(43);`

PROGRAMACIÓN IMPERATIVA EN JAVASCRIPT



PROGRAMA

1. ENTORNO DE PROGRAMACIÓN
2. PROGRAMAR EN EL EDITOR
3. VARIABLES, CONSTANTES Y SUS TIPOS
4. OPERACIONES BÁSICAS
5. CONDICIONALES
6. CICLOS
7. FUNCIONES





REPASO DE LA CLASE ANTERIOR

1. Repaso y aclaración de los tipos de variables vistos
2. Nuevos tipos: undefined, null, Symbol
3. Repaso y aclaración de arreglos
4. Operadores aritméticos
5. Operadores de incrementos y decremento
6. Operadores de asignación
7. Operadores de comparación
8. Métodos y funciones para el manejo de strings
9. Manejo y combinación de strings y numbers

MANEJO DE ARREGLOS

- **pop** => `let popArray = [1, 2, 3];`
 `popArray.pop();`
- **push** => `let pushArray = [1, 2, 3];`
 `pushArray.push(4);`
- **splice** => `let spliceArray = [1, 2, 3];`
 `spliceArray.splice(1, 1, "x", "y");`
- **reverse** => `[1, 2, 3].reverse();`

MANEJO DE ARREGLOS

- shift => `let shiftArray = [1, 2, 3];`
`shiftArray.shift();`
- unshift => `let unshiftArray = [1, 2, 3];`
`unshiftArray.unshift(0);`
- join => `["hello", "world"].join(",");`
- forEach => `[1, 2, 3].forEach(elemento => console.log(elemento));`

COMUNES EN STRINGS Y ARREGLOS

- map

- `["a", "b", "c"].map(e => e + "_JS");`
- `[1, 2, 3].map(elemento => elemento * 2);`

- filter

- `["aa", "bb", "ac"].filter(e => e.startsWith("a"));`
- `[1, 2, 3, 4].filter(elemento => elemento % 2 === 0);`

- reduce

- `["a", "b", "c"].reduce((a, e) => a + e, "_____");`
- `[1, 2, 3, 4].reduce((a, e) => a + e, 0);`

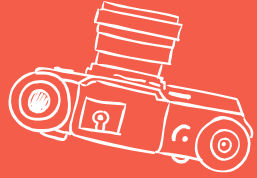
COMUNES EN STRINGS Y ARREGLOS

- length
 - `'Hola'.length;`
 - `[1, 2, 3].length;`
- concat
 - `'Hola'.concat(' Mundo')` ó `'Hola' + ' Mundo';`
 - `[1, 2].concat([3, 4]);`
- indexOf
 - `'Hola Mundo'.indexOf('Mundo');`
 - `[1, 2, 3].indexOf(2);`

COMUNES EN STRINGS Y ARREGLOS

- slice
 - `'Hola Mundo'.slice(5, 10);`
 - `[1, 2, 3, 4, 5, 6].slice(0, 3);`
- includes
 - `"javascript".includes("a");`
 - `[1, 2, 3].includes(3);`

PROGRAMACIÓN IMPERATIVA EN JAVASCRIPT



PROGRAMA

1. ENTORNO DE PROGRAMACIÓN
2. PROGRAMAR EN EL EDITOR
3. VARIABLES, CONSTANTES Y SUS TIPOS
4. OPERACIONES BÁSICAS
5. CONDICIONALES
6. CICLOS
7. FUNCIONES



ALGUNAS COSAS ANTES DE EMPEZAR

- Encuesta: <https://forms.gle/7R6WoZBB366K82gy8>
- Practico N1:
<https://sites.google.com/unc.edu.ar/argentinaprograma/practicos>



REPASO DE LA CLASE ANTERIOR

1. Métodos para manejo de strings
 - pop, push, splice, reverse
 - shift, unshift, join, forEach
2. Comunes en strings y arreglos
 - map, filter, reduce, length
 - concat, indexOf, slice, includes

FUNCIONES EN JAVASCRIPT

- `console.log("hello world");`
 - formatters: `%s`, `%i`, ...
- `JSON.stringify`
- `typeof 55;` `typeof "javascript";`
- `Math.random();`
- `Math.floor(3.44);`
- `Math.ceil(3.66);`
- `Math.round(3.44);` `Math.round(3.66);`