PROGRAMACIÓN WEB
teoría/práctica

GUÍA BÁSICA DE JAVASCRIPT, parte I

Guía Básica de Javascript, parte I

Complejidad de esta currícula: intermedia/alta
Tiempo aproximado de desarrollo: de (2-4) encuentros.

Qué necesitás saber?

- HTML Básico
- Conceptos Básicos de Programación (currículas Introductorias)

Temario de esta currícula

1. Qué es Javascript. Para qué y cómo se utiliza, ejemplos.
2. Uso de ALERT. Uso de PROMPT.
3. Sintaxis de Javascript.
4. Variables.
5. Estructuras de programación: condicional.
6. Operadores.
7. Arrays.
8. Estructuras de programación: for.

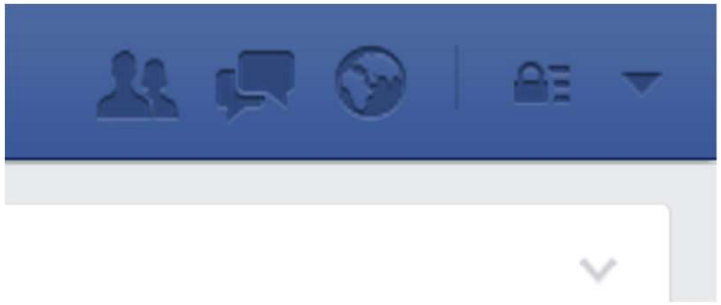
Guía de Javascript

- Te diste cuenta alguna vez que en algunos sitios webs, los elementos se mueven? Cambian de lugar, de color…. Te aparecen notificaciones, "cartelitos", este movimiento lo logramos usando **Javascript**!

- Preparate, porque en esta currícula vamos a aprender a programar con Javascript y a utilizarlo para darle vida a nuestro sitio.
- Este es un lenguaje que también se escribe en un editor de texto al igual que HTML y CSS. Para ejecutarlo solo hace falta integrarlo a un archivo HTML, muy parecido a como integrábamos CSS.
- Para ver los cambios que vayamos haciendo con Javascript solo debemos ejecutar en el navegador el archivo HTML.
- Una forma corta de llamar a Javascript es llamarlo "Js"

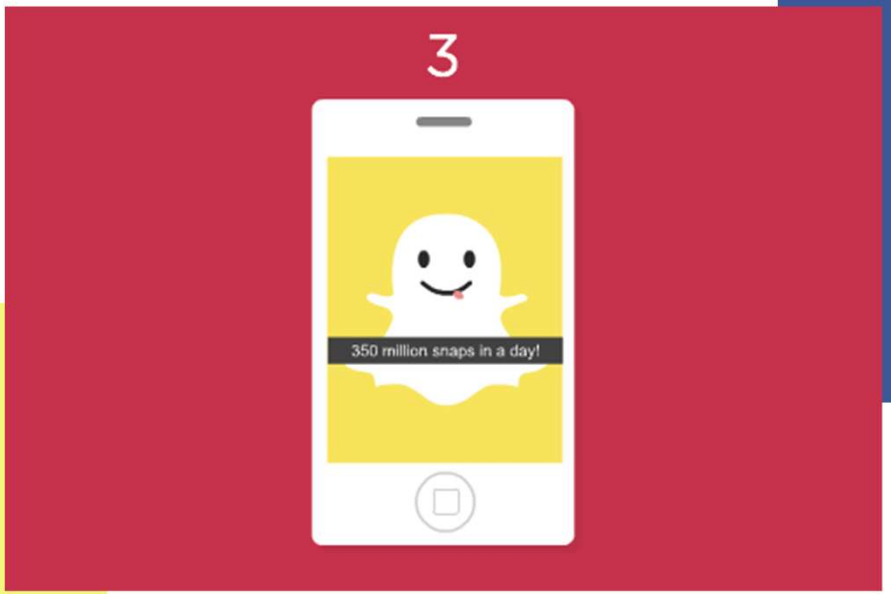
Qué es Javascript?

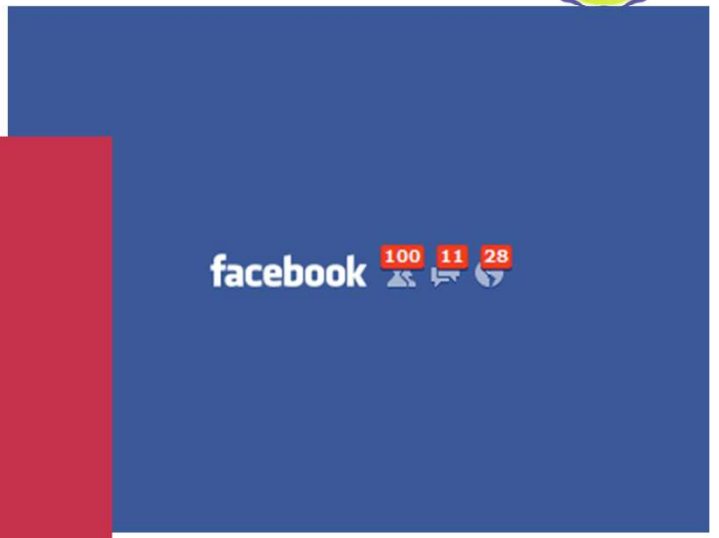
- Es un lenguaje de programación que se ejecuta dentro de nuestro navegador (recordemos que esto se llama del lado del **"cliente"**), y se utiliza en los sitios web
- Es un lenguaje muy rápido y muy extensamente utilizado.
- Javascript logra que los elementos HTML tengan dinamismo, esto quiere decir que cambien de color, forma o posición.
- También es posible hacer otro tipo de operaciones como por ejemplo trabajar con: cálculos, caracteres, palabras, etc.



Ejemplos del uso de Javascript

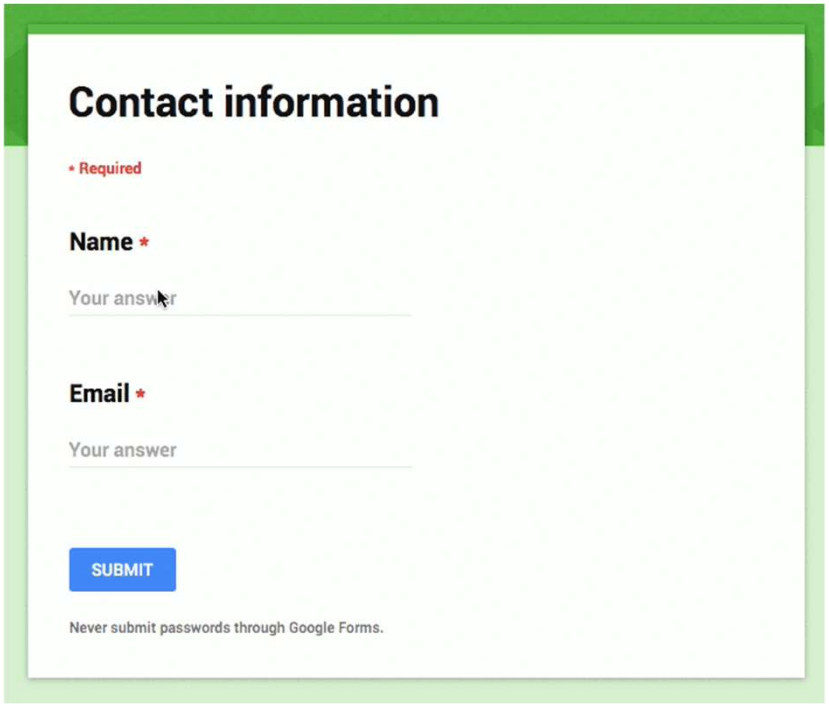
- Facebook
- Twitter
- Snapchat

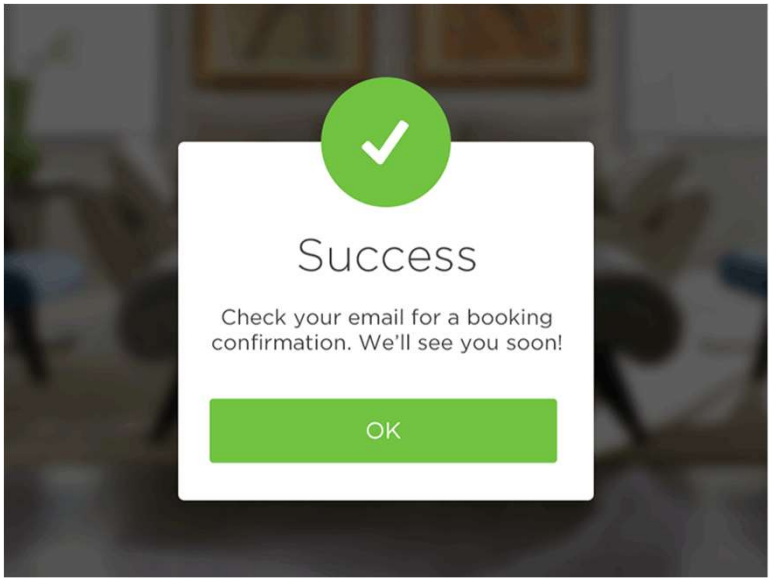




Ejemplos del uso de Javascript

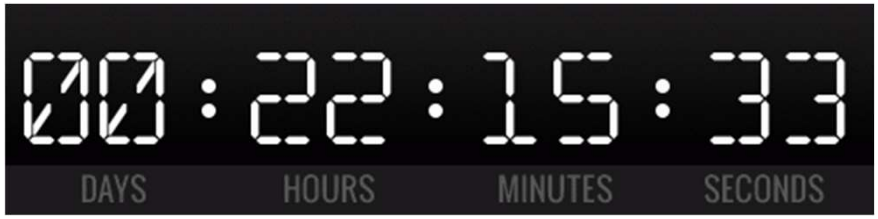
- Pop ups que nos saltan en distintos sitios web
- Cuando nos equivocamos en un formulario o hacemos algo bien





Ejemplos del uso de Javascript

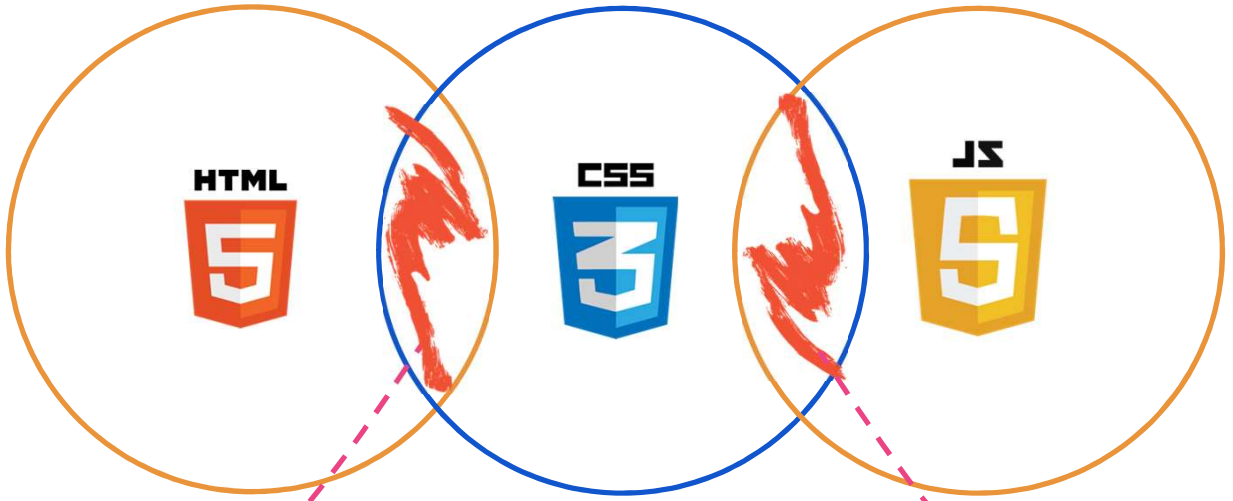
- Animaciones
- Juegos
- Y muchas cosas más!



Javascript

- Antes de empezar con este lenguaje, tenemos que saber que Javascript, HTML y CSS, son como hermanos! Siempre están juntos!
- Y así como los hermanos en la vida real, hay cosas que entre ellos comparten y hay otras en las que son muy diferentes
- En que son similares? Hay acciones dentro de una página web que las puede hacer HTML pero también CSS, por ejemplo: es posible achicar una imagen con HTML..aunque lo ideal es hacerlo con CSS!
- También hay acciones que pueden ser hechas con CSS, pero que también es posible hacerlas con Javascript

Javascript



En común con CSS

En común con Js

Cómo se utiliza Javascript?

- Al igual que con CSS hay 3 formas para incluir Javascript dentro de un documento de HTML. Esto es fundamental para que pueda funcionar

- Hay tres formas de escribir Javascript:

    - 1. Dentro del mismo archivo HTML
    - 2. En un archivo js aparte
    - 3. Dentro de la misma etiqueta HTML

- En general se utiliza tener un archivo aparte, pero también es muy común ver cualquiera de las otras dos prácticas, en especial si la cantidad de código de Js no es mucha.

1. Js dentro del mismo archivo HTML

- Se utiliza cuando no hay mucho código Js
- Dentro del archivo HTML se incrusta el código Js
- Esto es posible gracias a la etiqueta **<script></script>**
- **Veamos un pedacito de código:**

```
<body>
      <script>
            document.getElementById("demo").innerHTML = "Hola Mundo!";
      </script>
</body>
```

- **No importa si todavía no entendemos qué significa! :D**

2. Js externo a un archivo HTML

- Crear un archivo aparte (o varios) es la práctica más común cuando se tiene mucho código Js
- Desde los archivos HTML "llamaremos" a los archivos Js
- Ejemplo: Nuestro archivo index.html contiene una llamada al archivo "index.js"

```
<!DOCTYPE html>
<html>
     <body>
          <script src="index.js"></script>
     </body>
</html>
```

Nosotras usaremos los 3 métodos para hacer nuestras páginas web!

3. Js dentro de una etiqueta de HTML

- Este método se usa cuando la acción que queremos asociar a un elemento es muy puntual. Por ejemplo: Que al pulsar un botón de nuestra página web, se dispare un aviso

```
<!DOCTYPE html>
<html>
   <body>

   <h1>My First JavaScript</h1>

   <button type="button" onclick="document.getElementById('demo').innerHTML = Date()">Click me to display Date and Time.</button>

   <p id="demo"></p>

   </body>
   </html>
```

- Notemos que se dispara desde el atributo "onClick", es decir, cuando se hace click sobre el botón

No importa si todavía no entendemos qué significa! :D

1er Paso: Incluímos el archivo Js

- 1. Vamos a trabajar con la web que ya creamos en otras currículas. Por ahora empezaremos con el método 2, que es el de incluir un archivo externo.

- 2. Dentro de **<head></head>** o de **<body></body>** vamos a agregar la etiqueta:

```
<script src="index.js"></script>
```

Si tuviéramos más archivos Js solamente deberíamos agregar más sentencias como esa, por ejemplo:

```
<script src="index.js"></script>
<script src="script.js"></script>
<script src="otroscript.js"></script>
```

- Esta etiqueta lo que hace es "llamar" al archivo Js que vamos a crear. Lo llamamos según su nombre y su ubicación… En este caso el archivo index.js estará dentro de la carpeta "MiWeb", junto a el archivo index.html y junto a los archivos css.

Sintaxis de Javascript

- Cuando aprendemos un lenguaje, por ejemplo, inglés… tenemos que aprender ciertas estructuras para poder hablarlo y escribirlo. Si nos salimos de esas estructuras, probablemente no se entienda que estamos hablando inglés.

- Con los lenguajes de programación ocurre lo mismo! Cada lenguaje tiene definido la forma en que se debe "hablar"... solo que en este caso, si hablamos mal… la única que no va a entendernos es la compu!

- Sintaxis, que palabra rara! Sintaxis es la forma en que escribimos un lenguaje. Es un conjunto de reglas que deben seguirse para poder escribir el código. Cada lenguaje de programación tiene su propia sintaxis

Sintaxis de Javascript

- Normas básicas:
  - 1. No se tienen en cuenta los espacios en blanco y las nuevas líneas (Javascript los ignora)
  - 2. Se distinguen mayúsculas de minúsculas. Si se llegaran a intercambiar las mayúsculas y minúsculas el script no funcionaría.
  - 3. No se define el tipo de las variables. Al crear una variable no se define de que tipo es (por ejemplo, no se define que 3 es un número entero)
  - 4. No es necesario terminar cada sentencia con un punto y coma (;), en Javascript no es obligatorio.
  - 5. Se pueden incluir comentarios.

Variables en Javascript

- Javascript es un lenguaje de programación y como tal hace uso de: variables y estructuras lógicas (ver currículas introductorias)

- Recordemos que una variable es un lugarcito en memoria donde podemos guardar un tipo de dato (por ejemplo: número, caracter, palabra, etc)

- Como la podemos representar en Js, simplemente anteponiendo var cuando se crea la variable, luego no es necesario:

var nombre_variable = dato/operación que contiene

- Por ejemplo:

var suma = 9 + 2;

- En Js no hace falta siempre después de una sentencia poner el punto y coma, pero es conveniente para mantener cierto orden

Variables en Javascript

- También es posible dentro de una variable poner otras:

```
var operando1 = 9;
var operando2 = 2;
var suma = operando1 + operando2;
```

- Las variables solo pueden estar formadas por letras, números y símbolos (pero no pueden empezar con un número). Por ejemplo:

```
var $numero1;
var numero1;
```

- Las variables no solamente pueden contener números, también caracteres, palabras y hasta verdadero o falso! Ejemplos:

```
var minombre = "alejandra";
var mensaje = true;
```

Estructuras de Programación: Condicional

- Sabemos que programar es decidir y para poder tomar una decisión es necesario primero informarnos y luego decidir.

- En programación usamos mucho este tipo de estructura y se representa así en Js:

```
If (condición a evaluar) {
   // se toma la decisión
}
```

- Pensemos con un ejemplo: a que hora nos levantamos para ir al cole?

```
If (hora == 7) {
   mensaje = "es hora de levantarse para ir al cole!";
   alert(mensaje);
}
```

- Y a esto lo podemos leer: Si "la hora es 7am" entonces mostrar mensaje hay que levantarse!

2do Paso: Primer Hola Mundo

- 1. Ya creamos nuestro "index.js" ahora vamos a incorporar un aviso así cuando la persona ingresa al sitio web recibe un mensaje de bienvenida.
- Agregamos: alert("Bienvenido a mi sitio");
- Notá que el texto que queremos que se muestre en pantalla siempre tiene que salir entre comillas. **Alert** es una función de Javascript para escribir en pantalla.
- 2. Ahora crearemos una variable que tenga el valor "true".
- 3. Vamos a "preguntar", utilizando la estructura condicional que aprendimos, si el valor de la variable anterior es verdadera (por más de que ya sabemos que así sea). En caso de que sea verdadero, mostraremos otro mensaje por pantalla.

Estructuras de Programación: Condicional

- Sigamos pensando en el ejemplo de la alarma para ir al cole. Si la hora que indica el reloj es igual a las 7am, entonces me tengo que levantar...y si no es esa hora, entonces puedo seguir durmiendo!

- Si lo pensamos como programadoras:

```
If (hora == 7) {
        mensaje = "es hora de levantarse para ir al cole!";
}else{
    mensaje = "sigo durmiendo!";
}
alert(mensaje);
```

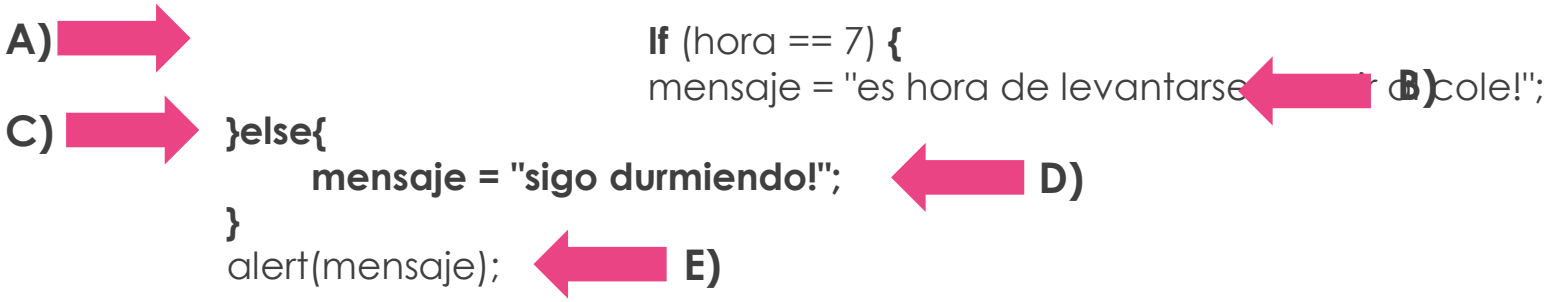
Si se cumple la condición, se ejecuta **SOLAMENTE** esta línea

Si **NO** se cumple la condición, se ejecuta la otra línea

El contenido del mensaje va a depender de que condición se cumpla, pero SIEMPRE se va a mostrar (porque no está dentro de ninguna condición)

Estructuras de Programación: Condicional

- Supongamos que la variable **hora** almacena la hora que se obtiene del sistema operativo. Observemos como la máquina iría ejecutando estas instrucciones:



- Se evalúa si la variable **hora** es igual a 7 (esa es la condición).
  - Si la hora es igual a 7 se ejecuta B) : se guarda la frase en la variable "mensaje".
  - Si la hora es distinta de 7 se ejecuta C) que solamente es un pasito antes de que se ejecute D).
- Por último, se muestra lo que tenga almacenado la variable "mensaje" que puede ser el mensaje B) o el mensaje D).

Estructuras de Programación: Condicional

- Notaste que si queremos mostrar el valor de una variable (en este caso la variable mensaje), no hace falta que pongamos comillas?
- Las comillas, son solo necesarias cuando queremos que cierto texto salga en la pantalla. Podemos combinar texto con el valor de las variables también
- Por ejemplo:

```
alert( mensaje, " ya son las: " , hora );
```

variable variable

- Y en pantalla se vería (si la hora fuera 7):

Es hora de levantarse para ir al cole!, ya son las: 7

Operadores de comparación

- Así como pudimos evaluar si (hora == 7) también podemos evaluar con otros operadores de comparación, los más frecuentes son:

Operador	Descripción
==	Equivalente
!=	No Equivalente
>	Mayor a
<	Menor a
>=	Mayor o igual
<=	Menor o igual

Un poco más allá...

- Reformulemos nuestra historia de la alarma para ir al cole: "**Si** *son las 7* **entonces** tengo que levantarme para ir al cole, **sino, si** *todavía no son las 7* entonces sigo durmiendo...y sino me quedé dormida!"

- Si pensamos como programadoras:

```
If (hora == 7) {
      mensaje = "es hora de levantarse para ir al cole!";
}else if (hora < 7) {
      mensaje = "sigo durmiendo!";
}else{
            mensaje = "me quedé dormida!";
}
alert(mensaje);
```

- Notemos que es posible seguir encadenando condiciones y seguir evaluando condiciones, todo va a depender de lo que tengamos que hacer

Operadores lógicos

- Volvamos a nuestra historia, esta vez para agregar más detalles: "**Si** *son las 7* ***y es lunes***...**entonces** tengo que levantarme para ir al cole porque tengo clase de música, **sino, si** *todavía no son las 7* entonces sigo durmiendo...y sino me quedé dormida!

```
If (hora == 7) && (dia == "lunes") {
      mensaje = "es hora de levantarse para ir al cole!";
}else if (hora < 7) {
      mensaje = "sigo durmiendo!";
}else{
            mensaje = "me quedé dormida!";
}
alert(mensaje);
```

- Lo que hicimos fue agregar otra variable (dia) y otra condición a evaluar (dia == "lunes")

Operadores lógicos

- Para eso están los operadores lógicos…Para agregar condiciones cuando sea necesario
- El operador que utilizamos recién se llama AND y simboliza una "y"
- Si queremos simbolizar una "o" entonces usamos OR, para situaciones donde se tenga que cumplir una condición o la otra
- Y si pensamos en hacer preguntas no solo en positivo? Por ejemplo:
  - Si la hora es distinta a 7
  - O si el día es distinto de lunes
- Para esto usamos el operador "negación".

Operadores lógicos

Operador	Ejemplo	Explicación
&&	If ((edad == 15) && (dia == lunes))	Si la hora es 7 y el día es lunes
||	If ((dia == nublado) || (dia == lluvioso))	Si el día está nublado o el día está lluvioso
!	If (! mostrarmensaje)	Si mostrarmensaje es falso

- Los paréntesis son muy importantes cuando vamos a usar un operador lógico. Aprendamos a utilizarlos :)

3er Paso: If, else y operadores

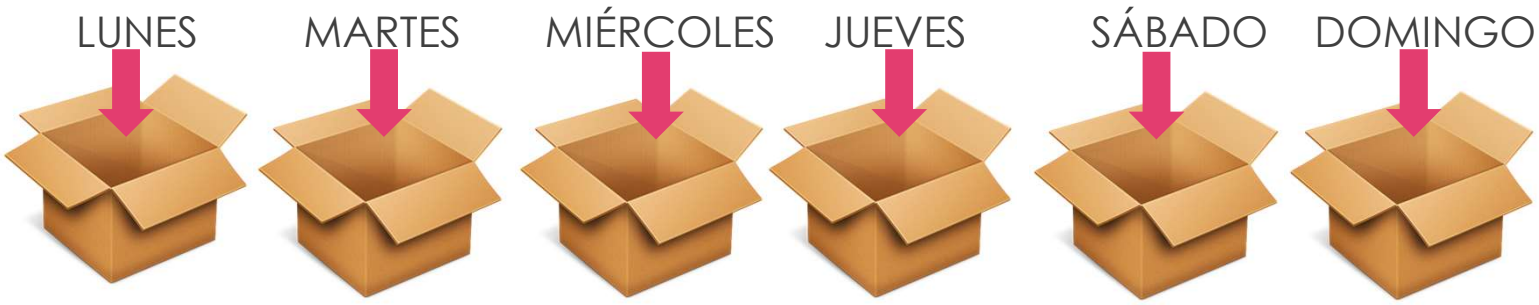
- 1. Vamos a utilizar otra función de Javascript además de "alert". Se llama "prompt" y se utiliza para que el usuario ingrese un dato. Escribimos:

```
var edad = prompt("Ingresá tu edad");
```

- 2. Lo siguiente que haremos es consultar una vez ingresado el dato, si "la edad es menor o igual a 14 y mayor o igual a 10" entonces mostrar un mensaje. También tenemos que preguntar "si la edad es mayor a 14, entonces debemos mostrar otro mensaje"...y por último si la edad es menor a 10, mostrar otro mensaje!

Arreglos (Arrays)

- Otra estructura muy importante son los arreglos. Imaginemos que tenemos muchas cajas... Y que en esas cajas podemos guardar muchas cosas (como las variables).
- La diferencia con una variable es que un arreglo es un conjunto de variables...y tendrá toda la información junta, en un mismo lugar..pero en diferentes "cajas"
- Por ejemplo:



Arreglos (Arrays)

- Para poder utilizar arreglos es necesario que tengan un nombre.
- Cada elemento que contenga el arreglo será tratado como un elemento que está ubicado en cierta posición.
- Ejemplo:

```
var dias_semana = [ "lunes", "martes", ",miercoles", "jueves", "viernes", "sabado","domingo ];
```

- Podríamos hacer lo mismo utilizando solamente variables, pero el código se volvería confuso e ineficiente:

```
var lunes = "lunes";
var martes = "martes";
.....
var "domingo" = "domingo";
```

Estructuras de Programación: For

- Cómo podríamos hacer para mostrar por pantalla todos los elementos de un arreglo? O trabajar con el contenido del arreglo? O simplemente ejecutar de forma repetitiva la misma acción?

- La respuesta a estas preguntas es la estructura for (para).

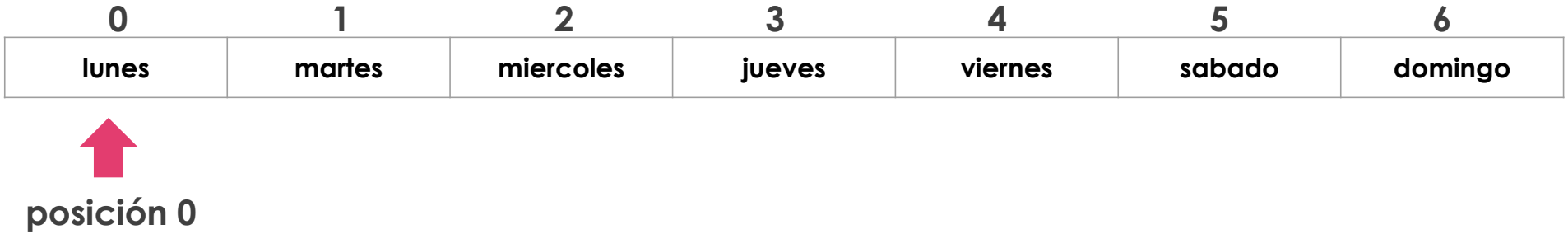
- Esta estructura permite que se pueda "recorrer" fácilmente un arreglo.

```
for( var i = 0; i < 7; i++ ) {
        alert(dias_semana[i];
}
```

- Es una estructura más compleja que un if así que vamos a estudiarla por partes..

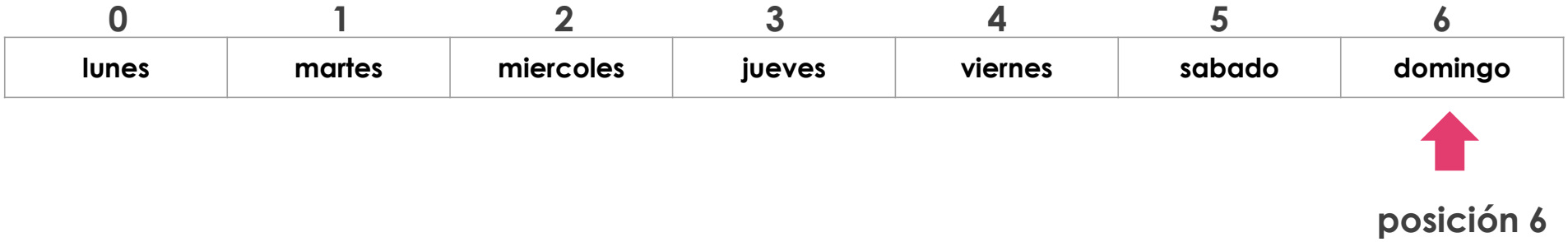
Estructuras de Programación: For

- Siempre que se va a recorrer un arreglo se debe poner desde que posición se empieza a recorrer. Por eso es que tenemos:
- for(var i = 0; i < 6; i++) {
- Allí estamos declarando una variable i que empieza en 0. Esto quiere decir que queremos empezar a recorrer el arreglo DESDE LA POSICIÓN 0.



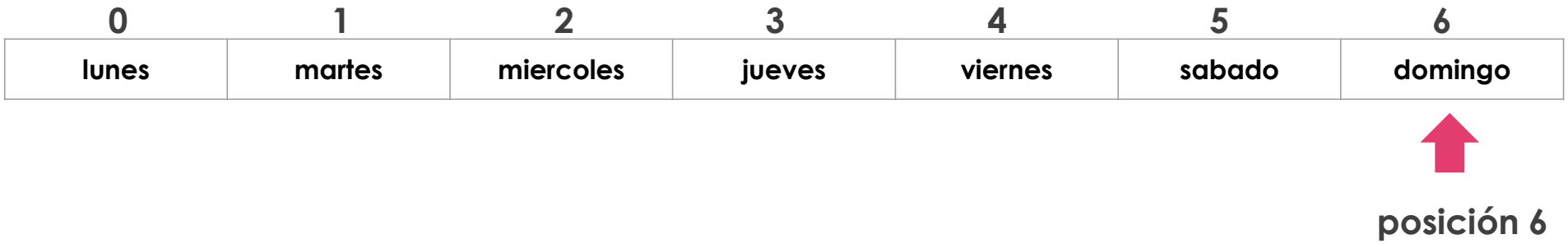
Estructuras de Programación: For

- Luego decimos que queremos que se recorra hasta la posición 6 nada más (ya que en total son 7 elementos)
- for(var i = 0; i < 7; i++) {
- La variable i se irá incrementando hasta llegar a 6...



Estructuras de Programación: For

- Y la última parte, significa que la variable i se debe incrementar de a un número.
- for(var i = 0; i < 6; i++) {



Estructuras de Programación: For

- La dinámica del recorrido es la siguiente:

```
for( var i = 0; i < 7; i++ ) {
        alert(dias_semana[i];
}
```

1. La variable i empieza en 0
2. Luego la máquina pregunta, es i menor a 7? Como i == 0, sigue
3. Muestra por pantalla lo que hay en el array dias_semana en la posición i (que en este caso por ahora es 0), se muestra "lunes";
4. Como es un ciclo, vuelve al for, pero esta vez la variable i se incrementa en 1.
5. De nuevo la misma pregunta, es i menor a 7? Como i == 1, sigue
6. Muestra por pantalla lo que hay en el array, se muestra "martes"...

Así sucesivamente hasta que la i sea igual a 7 y se termina la recorrida!

¿Preguntas?

Desafío

- Te animás a agregar la estructura for dentro de tu sitio web?
- Probá agregar más estructuras condicionales y otras variables

MUCHAS GRACIAS