

Introducción a la programación

Clase 3

Jordi Collell

jordi@tempointeractiu.com

@galigan

Repaso sesión anterior

- > Cliente / Servidor
- > Variables
- > Scripts en Archivos
- > Flujo de Trabajo:
 Editar >> 🍏+S >> 🍏+R
- > Variables >> tipos de datos >> typeof()
- > // /* */
- > ==, ===, !=, !==, <=, >=, >, <
- > if / else if / else
- > ++, --, +=, -=, *=, /=

Repaso sesión anterior

```
// operadores lógicos
(a == b)
(a != b)
(a > b && a <= c)
(a >= b || b < c)
a == true
b == false
```

```
// condicionales
if(a=="un valor") {
    alert('la respuesta 1')
} else {
    alert('la respuesta 2')
}

if(a==b) {
    // bloque
} else if(a==c) {
    // bloque
} else {
    // bloque
}
```

Condicionales

```
a = prompt('num?');  
if(a<1) {  
    alert('1')  
} else if (a<2) {  
    alert('2')  
} else if (a<10) {  
    alert('10')  
} else {  
    alert('fin')  
}
```

> Un condicional es una instrucción que nos permite modificar el flujo lógico de un programa en función de condiciones lógicas.

> En las condiciones

```
if(a<1) {}
```

podemos usar expresiones lógicas del tipo:

- < mas pequeño que.
- > mas grande que.
- <= mas pequeño o igual a.
- >= mas grande o igual a
- == igual a
- != distinto a
- === igual en valor y tipo de dato
- !== distinto en valor y tipo de dato

Repeticiones for

```
for(i=0; i<50; i++) {  
    console.info('valor', i)  
}
```

- > El bloque de control, for, sirve para repetir una operación **n** veces.
- > El fragmento de código, generará 50 mensajes en la consola de javascript.
- > Existen mas bloques de control que repetiremos

Bloques switch

```
switch(vari)
{
    case "a":
        // bloque
        break
    case "b":
        // bloque
        brake
    default:
        // bloque
        break
}
```

> Un bloque **switch** es un bloque de código parecido al **if**, funciona de la misma forma, pero nos permite organizar el código de una forma más legible.

> Los case, nos permiten entrar casos, con la instrucción **break**, romperemos el flujo de ejecución.

> El operador **default** captura los casos no especificados por case.

Recordar:

Además de que un programa funcione, debe de ser legible. Otras personas sean capaces de entender lo que hemos escrito

ejercicio 7

Imaginemos que debemos programar la entrada organizada a un avión. El programa debe de recibir el número de fila y devolver, puerta delantera o trasera.

El avión tiene 30 filas. Hasta la fila 15 deben de entrar por puerta delantera. Hasta fila 30 por puerta trasera

ejercicio 8

Realizaremos una pequeña calculadora. En primer lugar nuestro programa nos solicitará un número. Seguidamente una operación y un tercer número.

Pista: Guardaremos los números en variables y deberemos de efectuar la operación realizando un condicional.

Esta vez, generaremos la salida en lugar de con un alert, usando la instrucción:

```
document.write(resultado)
```

que nos mostrará el resultado en el propio documento html en qué reside el código.

ejercicio 9

Realizaremos una pequeño programa que nos escriba en pantalla una série numérica, por ejemplo $i*i$

```
for(var... )
```

ejercicio 10

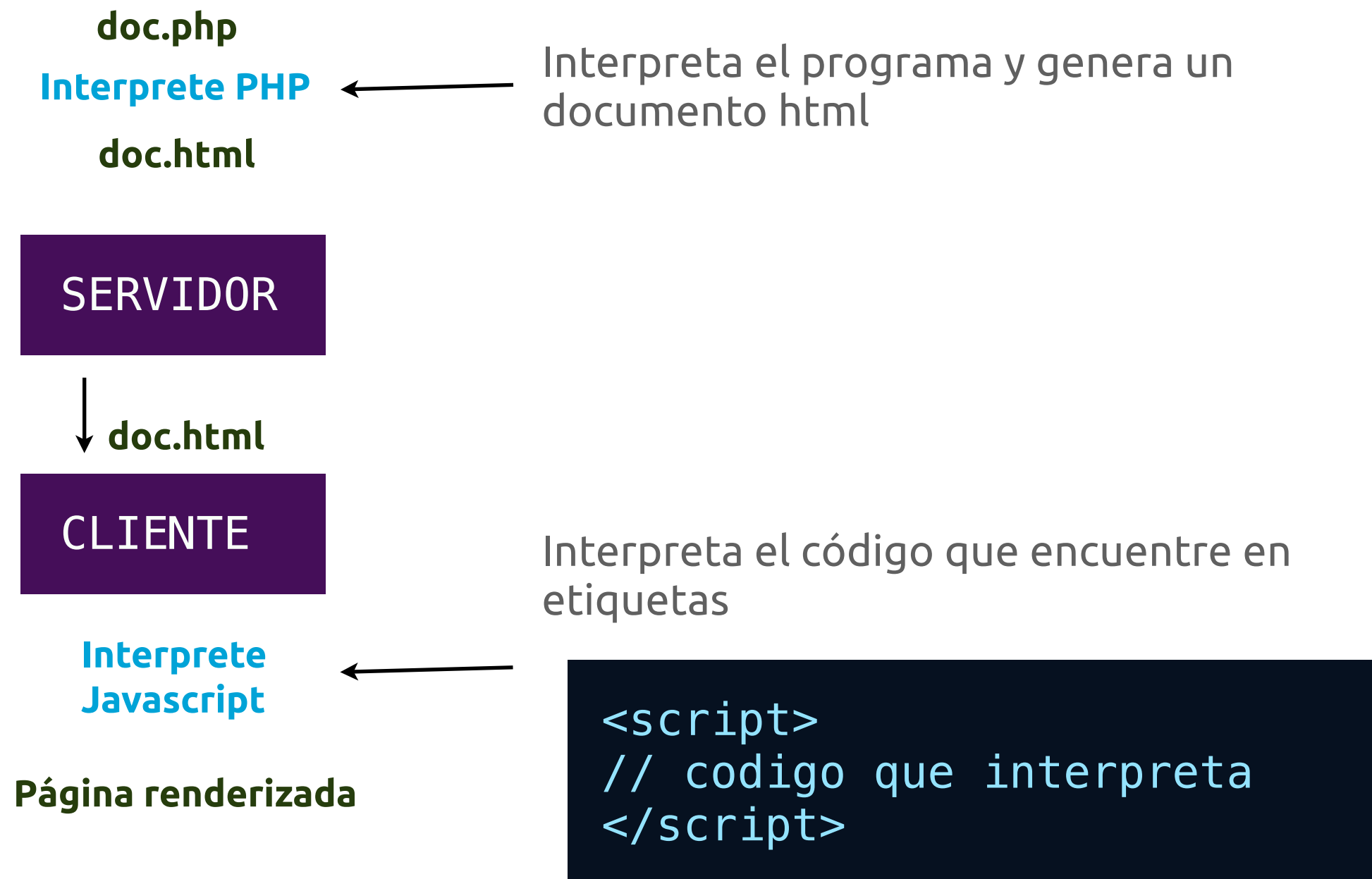
Ejercicio, calcula el factorial de 5, usando un ***for*** y sabiendo que el factorial se calcula con:

$$5*4*3*2*1$$

Serías capaz de generar un programa que calcule el factorial de **n**?

cliente vs servidor

> PHP vs Javascript: ¿Donde se ejecuta cada código?



clientes vs servidores

Podemos explorar algunos ejemplos de websites para ver las partes que se ejecutan en el cliente y las que se ejecutan en el servidor:

<http://meneame.net>

<http://pinterest.com>

<http://facebook.com>

Funciones

```
function saluda(name)
{
  alert("Hola "+name)
}

function suma(a, b) {
  return a+b
}
```

```
res = suma(5+10)
res == 15
```

- > Una función es un conjunto de instrucciones agrupado como una nueva instrucción.
- > Una función recibe unas variables a,b y puede o no devolver otro resultado.
- > El objeto de una función es reagrupar código que en el futuro reorganizaremos.
- > Una función nos resolverá un problema pequeño que podemos aislar del problema grande
- > El código de una función, no se ejecuta hasta que la llamamos.

Funciones

```
function saluda(name)
{
    alert("Hola "+name)
}

function suma(a, b) {
    return a+b
}
```

> Cuando una función devuelve algo, termina con un *return*, el resultado puede ser capturado.

```
resul = suma(5+10)
resul = resta(10-8)
```

Podemos definir directamente la función suma en la consola

```
function suma(v, v2) { return v+v2; }
```

ejercicio 11

Escribiremos una función que calcule un descuento, recibiendo dos parámetros, la cantidad sobre la que descontar y el porcentaje:

```
function descuenta(cantidad, porcentaje){  
  
}
```

La colocaremos en un ejercicio y la usaremos mediante la consola.

Modelo de eventos

¿Que es un evento?

- > Conceptualmente en programación un evento es un suceso que pasa desencadena una acción.
- > Prácticamente podemos asignar una función que **se ejecutará** cuando el evento se desencadene.

Modelo de eventos

```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.5.1/jquery.min.js"> </script>
<script type="text/javascript">

$(document).ready(function(){

    $('#boton').click(function(){
        alert( $('#campo').val() )
    }

})

</script>

</head>
<body>
<input type="text" name="valor" id="campo"/>
<input type="button" name="bt" value="Probar" id="boton"/>
</body>
</html>
```

<http://dl.dropbox.com/u/7625280/eventos.html>

ejercicio 12

Adaptaremos el ejercicio anterior, para que cuando pulsamos el botón nos aplique el descuento sobre el valor que contenga el campo.

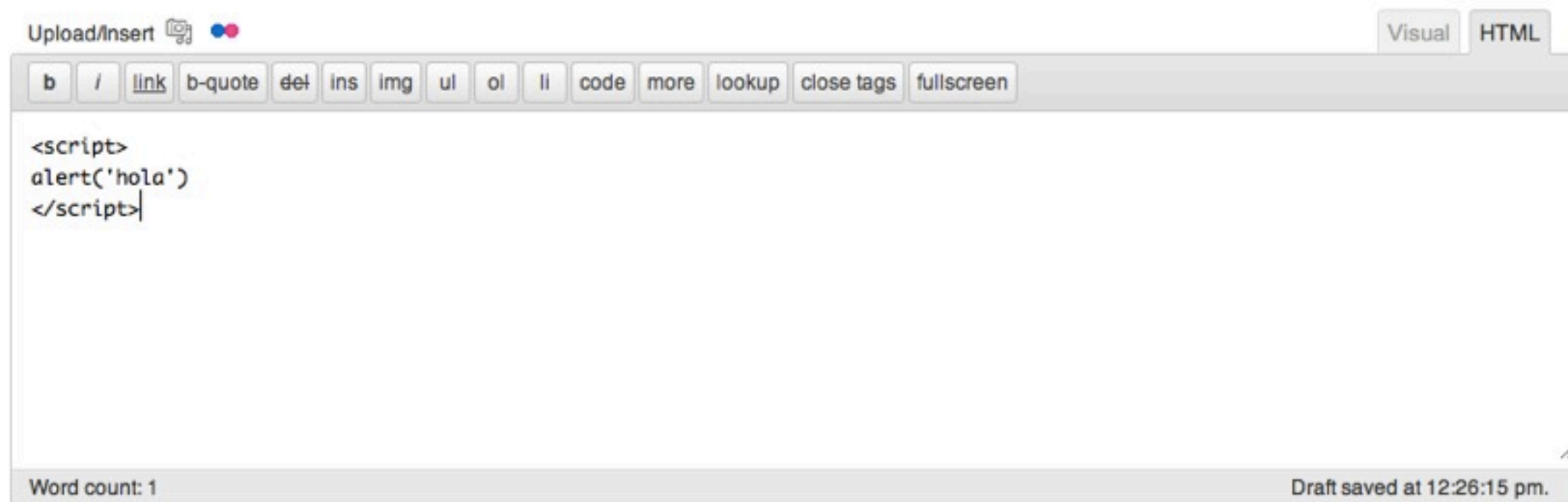
```
function descuentar(cantidad, porcentaje){  
  
}
```

Para empezar podemos utilizar la plantilla:

<http://dl.dropbox.com/u/7625280/plantilla.html>

ejercicio 13

Una vez realizado el ejercicio, lo publicaremos en el blog, editando el post en contenido html



Realizaremos una pequeña calculadora:

Calculadora en Javascript: