

PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
PENSAMIENTO VISUAL IDI1015

ESTEBAN SANDOVAL
EJSANDOVAL@UC.CL

AYUDANTÍA

javascript

PRESENTACIÓN DISEÑADA Y ELABORADA
POR ESTEBAN SANDOVAL

*BASADA EN “INTERACTIVE DATA VISUALIZATION FOR THE
WEB” DE SCOTT MURRAY



AYUDANTÍA 2

python

PROCESAMIENTO
Y LIMPIEZA DE
DATOS

AYUDANTÍA 3

html + css

ENTORNO WEB Y
DISEÑO

AYUDANTÍA 4

javascript

CONTENIDO
DINÁMICO E
INTERACCIÓN

AYUDANTÍA 5 Y 6

librerías.js

VISUALIZACIÓN
DE DATOS

javascript

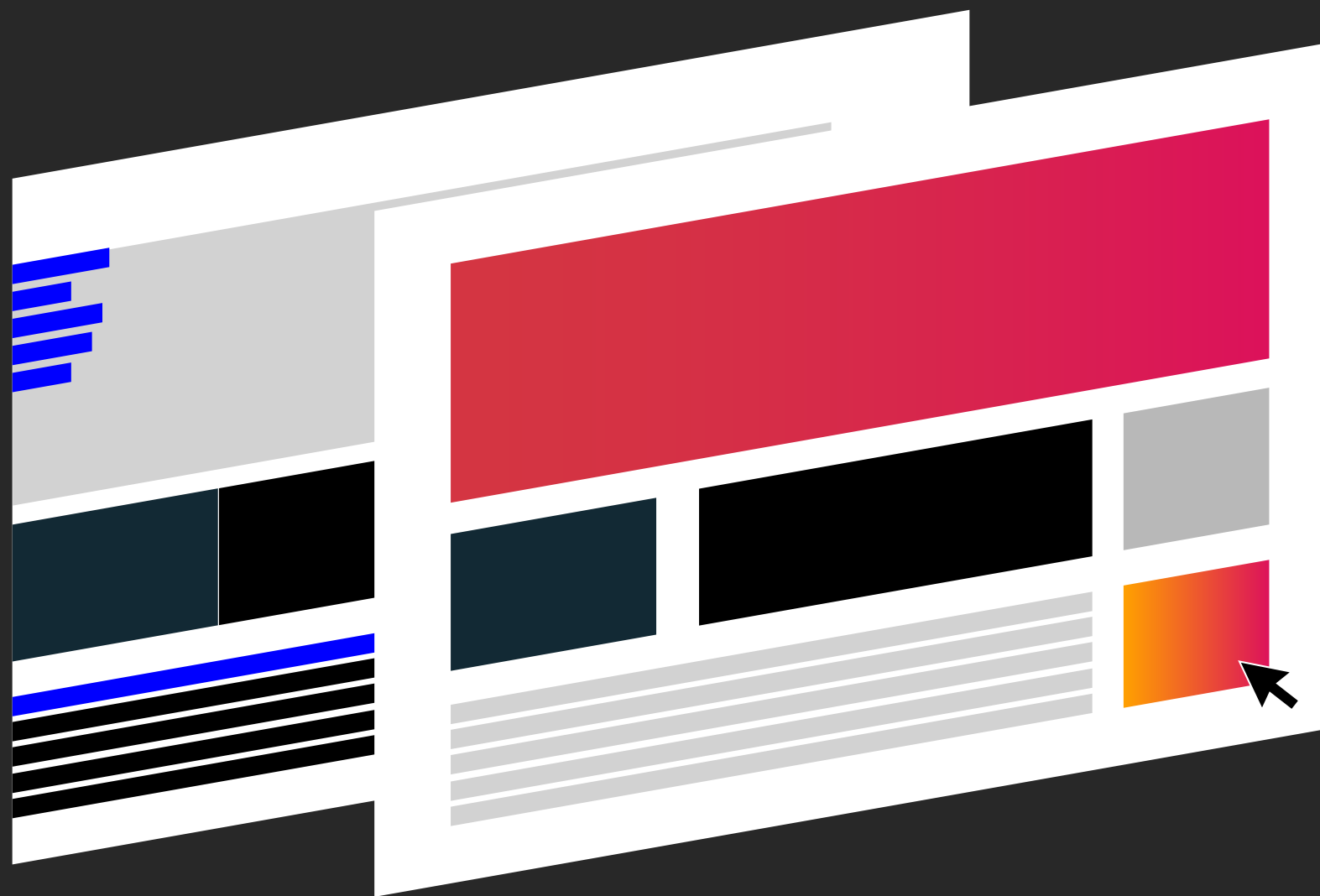
javascript











INTERACCIÓN
DINÁMICA

SIN TENER QUE *actualizar* POR CADA ACCIÓN

javascript

1 VARIABLES

2 ARREGLOS

3 OBJETOS

4 JSON

5 FUNCIONES

6 REFERENCIAS

7 JQUERY

1 VARIABLES

Las variables se definen de la siguiente forma

definición de variable

```
var ten = 10;  
console.log(ten*ten)
```

Impresión en consola

2 ARREGLOS

Estructura de datos para agrupar elementos de forma ordenada

```
var numbers = [ 5, 10, 15, 20, 25 ];  
var names = [ "Claudio", "Oswaldo", "Vanesa" ];  
numbers[2] //Returns 15
```

3 OBJETOS

Estructura de datos más compleja que los arreglos para poder almacenar y asociar distintos tipos de información.

```
var fruit = {  
  kind: "grape",  
  color: "purple",  
  quantity: 12,  
  tasty: true  
};
```

3 OBJETOS

Se puede referenciar cada uno de los atributos del objeto a través del “.”

```
fruit.kind //Returns “grape”  
fruit.color //Returns “red”  
fruit.quantity //Returns 12  
fruit.tasty //Returns true
```

3 OBJETOS

Podemos mezclar arreglos y objetos de manera libre.
Por ejemplo,

```
var fruits = [  
  {  
    kind: "grape",  
    color: "red",  
    quantity: 12,  
    tasty: true  
  },  
  {  
    kind: "kiwi",  
    color: "brown",  
    quantity: 98,  
    tasty: true  
  },  
  {  
    kind: "banana",  
    color: "yellow",  
    quantity: 0,  
    tasty: true  
  }  
];
```


3 OBJETOS

Luego realizamos llamadas “anidadas”.

```
fruits[0].kind == "grape"  
fruits[0].color == "red"  
fruits[0].quantity == 12  
fruits[0].tasty == true  
fruits[1].kind == "kiwi"  
fruits[1].color == "brown"  
fruits[1].quantity == 98  
fruits[1].tasty == true  
fruits[2].kind == "banana"  
fruits[2].color == "yellow"  
fruits[2].quantity == 0  
fruits[2].tasty == true
```

4 JSON

La única diferencia entre un objeto y un JSON es que los atributos también son strings.

```
var jsonFruit = {  
  "kind": "grape",  
  "color": "red",  
  "quantity": 12,  
  "tasty": true  
};
```

* CONDICIONES Y CONTROL DE FLUJO

Los comparadores son idénticos a los de Python.

==	igual que
!=	distinto/desigual
<	menor que
<=	menor o igual que
>	mayor que
>=	mayor o igual que

* CONDICIONES Y CONTROL DE FLUJO

La estructura general del if y else se mantiene similar a la de Python.

```
var edad = 23;  
var nombre = "Luis";  
  
if (edad < 18){  
    console.log(nombre + " es menor de edad");  
}  
else if (edad >= 21){  
    console.log(nombre + " es mayor de edad internacionalmente");  
}  
else {  
    console.log(nombre + " es mayor de edad");  
}
```

* CONDICIONES Y CONTROL DE FLUJO

Sin embargo, el for tiene algunos cambios que es necesario destacar:

```
for (initialization; test; update) {  
    //Code to run each time through the loop  
}  
  
for (var i = 0; i < 5; i++) {  
    console.log(i); //Prints value to console  
}
```

5 FUNCIONES

Las funciones se definen como variables.

```
var square = function(x) {  
    return (x * x);  
};
```

5 FUNCIONES

Ejemplo de una llamada de función:

```
var makeNoise = function( {  
    console.log("Pling!");  
});  
  
makeNoise();  
  
// En consola  
Pling!
```

6 REFERENCIAS



.html



.css



js



.js



.js

6 REFERENCIAS



.html



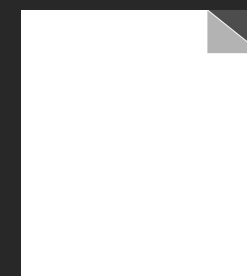
.css



js



.js



.js

6 REFERENCIAS

Se debe hacer referencia en el archivo .html en el cuál se quiere ejecutar el JavaScript o se puede incluir en el código.

```
<head>
  <title>Page Title</title>
  <script type="text/javascript" src="myscript.js"></script>
</head>

// o también...

<body>
  <script type="text/javascript">
    alert("Hello, world!");
  </script>
</body>
```

7 JQUERY

jQuery

7 JQUERY

Los selectores funcionan muy similar a los de CSS.

```
$(selector).accion()
```

7 JQUERY

Por ejemplo, para un elemento con id = "container".

```
$("#container").append( '<div>nuevo div</div>' );
```

7 JQUERY

Antes de todo archivo que utilice jQuery...

```
$(document).ready(function(){  
    ▪  
    ▪  
    ▪  
});
```

PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
PENSAMIENTO VISUAL IDI1015

CLAUDIO SCHEIHING CUEVAS
CNSCHEIHING@UC.CL

AYUDANTÍA

javascript

PRESENTACIÓN DISEÑADA Y ELABORADA
POR ESTEBAN SANDOVAL

