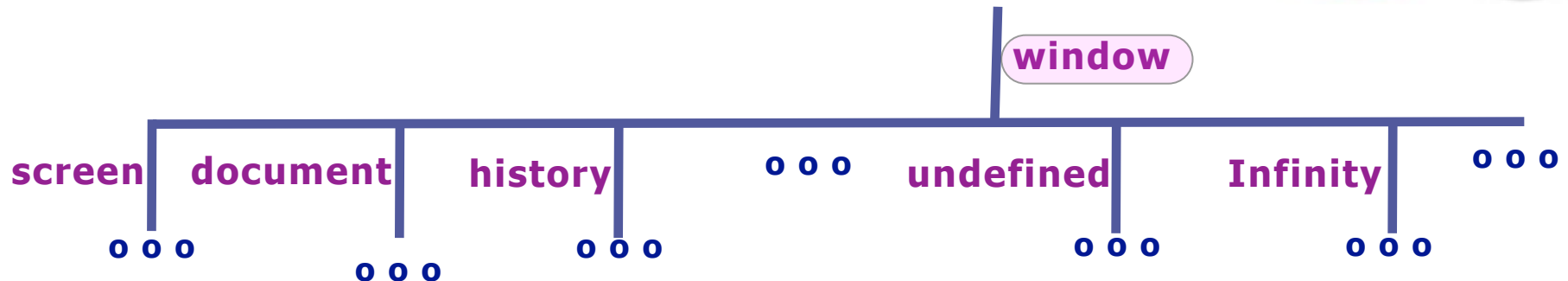




Tema 6.1

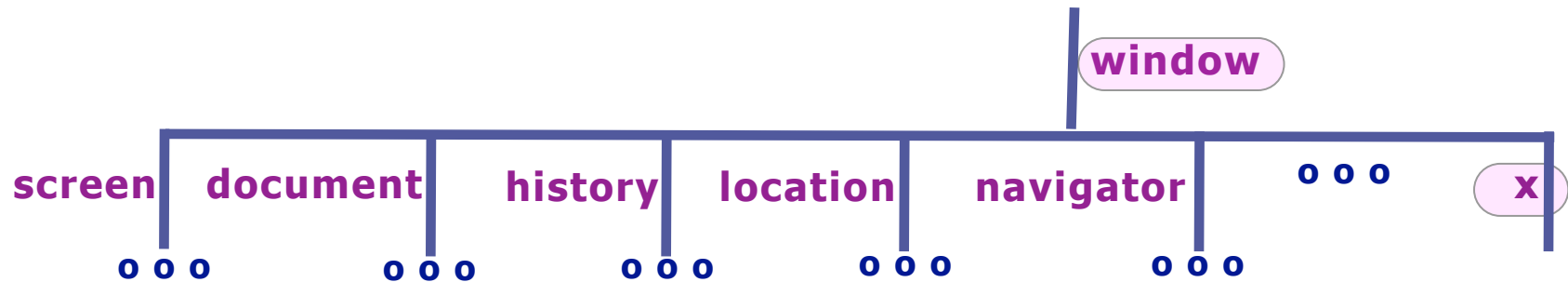
Javascript: DOM - Document Object Model

Objeto window

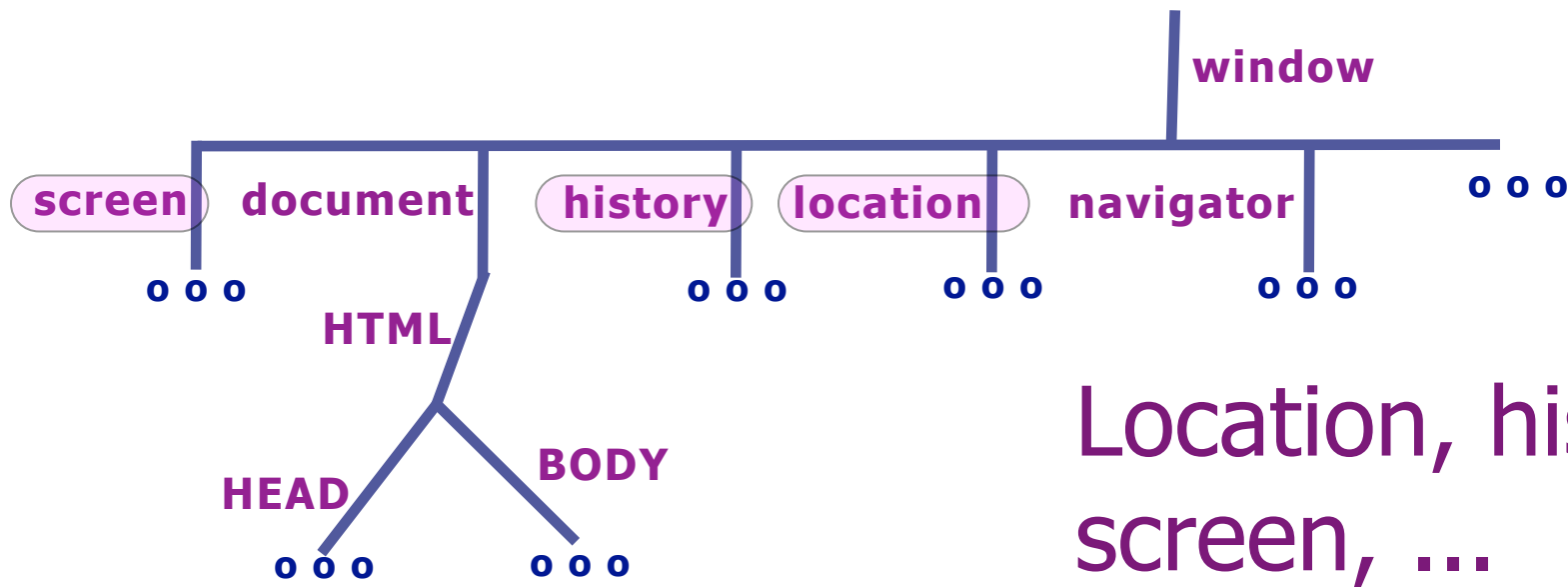


- ◆ JavaScript se ejecuta dentro del objeto global **window**
 - sus propiedades son los recursos de JavaScript:
 - ◆ **Navegador**: screen, document, history, location, navigator,
 - ◆ **Genéricos**: undefined, Infinity, NaN, isNaN(), ..., <constructores>, ...
 - **window** puede referenciarse también como **this** u omitirse
 - ◆ **window.screen** es equivalente a **this.screen** y a **screen**
- ◆ Mas información en: http://www.w3schools.com/jsref/obj_window.asp

Variables globales y el entorno de ejecución



- ◆ Un programa JavaScript se ejecuta con el objeto **window** como entorno
 - las propiedades de window son visibles en todo el programa
- ◆ Las **variables globales** de JavaScript son **propiedades de window**
 - **window.x = 1;** es equivalente a **this.x = 1;** o a **x = 1;**
 - ◆ **x = 1;** si x no existe, crea una nueva propiedad (variable global) de window



- ◆ **location:** propiedad que contiene el URL a la página en curso
 - `location = "http://www.upm.es"` // Carga la página en el navegador
 - ◆ **location.reload()** re-carga la página en curso
 - propiedades: **href** (url), **protocol**, **hostname**, **port**, **pathname**, **search** (query), ...
- ◆ **history:** propiedad con la historia de navegación
 - Métodos para navegar por la historia: **history.back()**, **history.forward()**, ...
- ◆ **screen:** dimensiones de la pantalla
 - **width**, **height**, **availWidth**, **availHeight**: para adaptar apps a pantallas móviles
- ◆ Documentación: <http://www.w3schools.com/jsref/>

window

```
<!DOCTYPE html>
<html><head><meta charset="UTF-8"></head>
<body>
<pre>
<script type="text/javascript">

    document.write("window.x      => ");
    document.writeln(window.x + "\n");

    x = 1;

    document.writeln("Se crea variabllle global x = 1; \n");

    document.writeln("        x      => " + x);
    document.writeln("    this.x    => " + this.x);
    document.writeln("window.x    => " + window.x + "\n");

    document.writeln("location.protocol => " + location.protocol);
    document.writeln("location      => \n" + location + "\n");

    document.writeln("screen.width  => " + screen.width);
    document.writeln("screen.height => " + screen.height);
</script>
</pre>
</body>
</html>
```

```
file:///Users/jq/Desktop/window.htm

window.x      => undefined

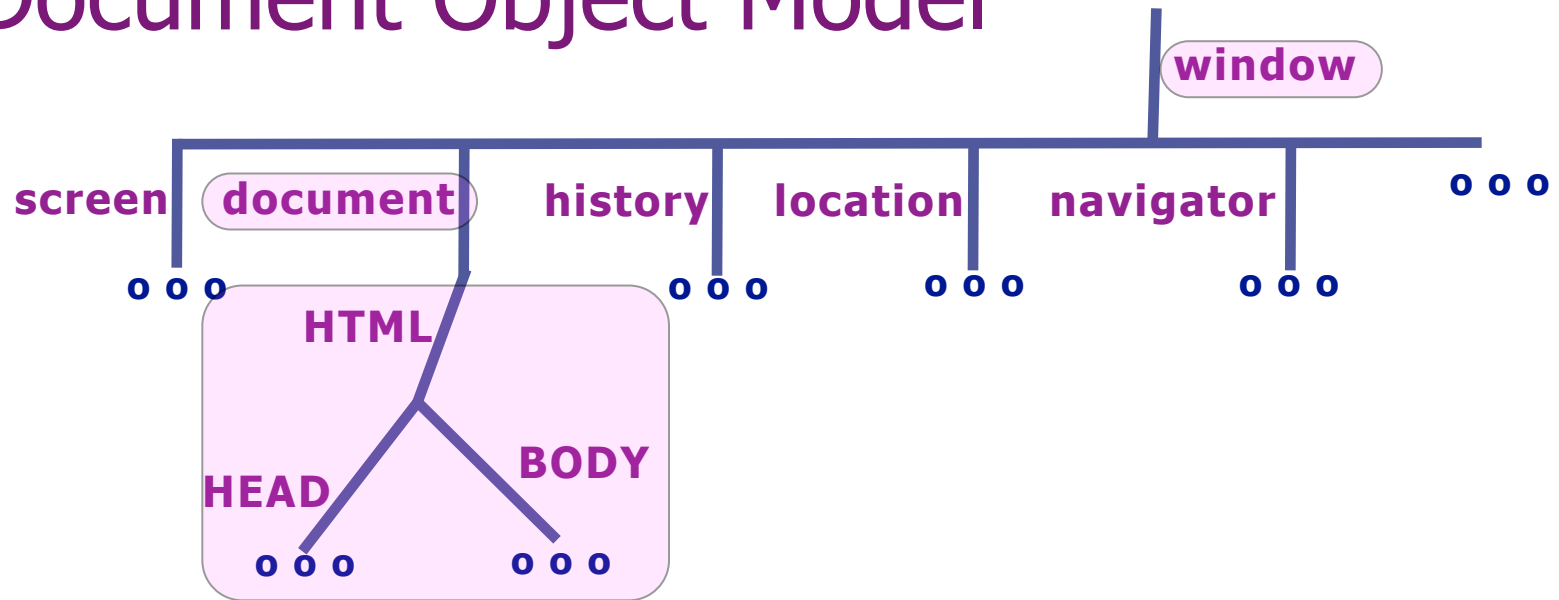
Se crea variabllle global x = 1;

        x      => 1
    this.x    => 1
window.x    => 1

location.protocol => file:
location      =>
    file:///Users/jq/Desktop/window.htm

screen.width  => 1920
screen.height => 1080
```

DOM: Document Object Model



- ◆ El objeto **window** tiene una propiedad **document** muy importante
 - **document** contiene el objeto con la página HTML cargada en ese momento
 - ◆ las propiedades de **document** contienen los elementos de la página
- ◆ El método **document.getElementById("id")** devuelve el objeto de document
 - asociado al elemento de la página HTML con **identificador "id"**
 - ◆ **Atributos "id=..."** son el interfaz entre **programa.js** y **página HTML**

- ◆ `getElementById("fecha")` devuelve un objeto DOM
 - en el ejemplo devuelve el objeto DOM asociado a `<div id="fecha"></div>`
 - ◆ **id** identifica unívocamente cada elemento
- ◆ Propiedad **innerHTML**
 - Permite extraer/insertar HTML en el elemento del documento

The image shows a code editor window titled '25-date_id.htm' and a web browser window titled 'Ejemplo fecha y hora'. The code editor contains the following HTML and JavaScript code:

```
<!DOCTYPE html>
<html>
<head>
<title>Ejemplo fecha y hora</title>
<meta charset="UTF-8">
</head>

<body>
<h2>La fecha y hora son:</h2>

<div id="fecha"></div>

<script type="text/javascript">
  var cl = document.getElementById("fecha");
  cl.innerHTML = new Date( );
  // Insertar Date en elem con id="clock"
</script>
</body>
</html>
```

The web browser window displays the rendered output:

La fecha y hora son:
Sun Sep 08 2013 12:46:50 GMT+0200 (CEST)

Blue dashed arrows indicate the flow of data: one arrow points from the `<div id="fecha"></div>` line in the code editor to the rendered text in the browser, and another arrow points from the `document.getElementById("fecha")` line in the JavaScript code to the same `<div id="fecha"></div>` line in the HTML code.

Acceso a DOM

© Juan Quemada, DIT, UPM

```
<!DOCTYPE html>
<html>
<head>
<title>DOM</title>
<meta charset="UTF-8">
</head>
<body>
```

```
<h2> Screen </h2>
```

```
    <!-- tabla con propiedades de screen -->
<table id="tabla"></table>
```

```
<script>
```

```
var tabla = document.getElementById("tabla");
var i=0;
```

```
for (i in screen)    //cada iteración genera una fila de la tabla
    tabla.innerHTML+="<tr><td>" + i + "</td><td> = " + screen[i] + "</td></tr>";
```

```
</script>
</body>
</html>
```

window.screen

Screen

availTop	= 22
width	= 2560
availHeight	= 1418
height	= 1440
availWidth	= 2512
availLeft	= 48
colorDepth	= 24
pixelDepth	= 24

Funciones de selección de elementos

◆ getElementById("my_id")

- Es el mas sencillo de utilizar porque devuelve un objeto o null
 - ◆ El objeto DOM con el identificador buscado
 - ¡Un identificador solo puede estar en un objeto de una página HTML!

◆ getElementByName("my_name"), getElementsByTagName("my_tag"), getElementsByClassName("my_class"), querySelectorAll("CSS selector"),...

- Devuelven una matriz de objetos
 - ◆ Por ejemplo: getElementByName("my_name")[0]
 - referencia el primer elemento con atributo **name="my_name"**

Ejercicio

- ◆ Si tenemos una página HTML con el siguiente contenido

```
<!DOCTYPE html>
<html> .... <body>
  <h4 id="id1" >Título</h4>

  <p id="id2">Texto</p>

  <div id="id3"></div>

  <script type="text/javascript">
    ..... // script con expresiones de abajo
  </script>
</body>
</html>
```

- ◆ Como se evaluarán las siguientes expresiones si estuviesen en el script

document.getElementById("id1").innerHTML	=>	undefined, "", "Título", "Texto"
document.getElementById("id2").innerHTML	=>	undefined, "", "Título", "Texto"
document.getElementById("id3").innerHTML	=>	undefined, "", "Título", "Texto"

Ejercicio

- ◆ Construir una página HTML
 - que liste todas las propiedades del objeto DOM **window**
 - ◆ que no sean objetos
- ◆ El script debe generar una tabla HTML con 2 columnas
 - La 1a columna contendrá los nombres de las propiedades de objeto
 - La 2a columna contendrá
 - ◆ el valor de cada propiedad de objeto, si no es un objeto (es un tipo)
 - ◆ el string “objeto?”, si el contenido es un objeto
- ◆ Se recomienda utilizar un script similar a los anteriores
 - que utilice la sentencia “for/in”
 - ◆ para extraer las propiedades e insertarlas en la página HTML con formato de tabla