

Fonaments de la programació

Identificadors i variables

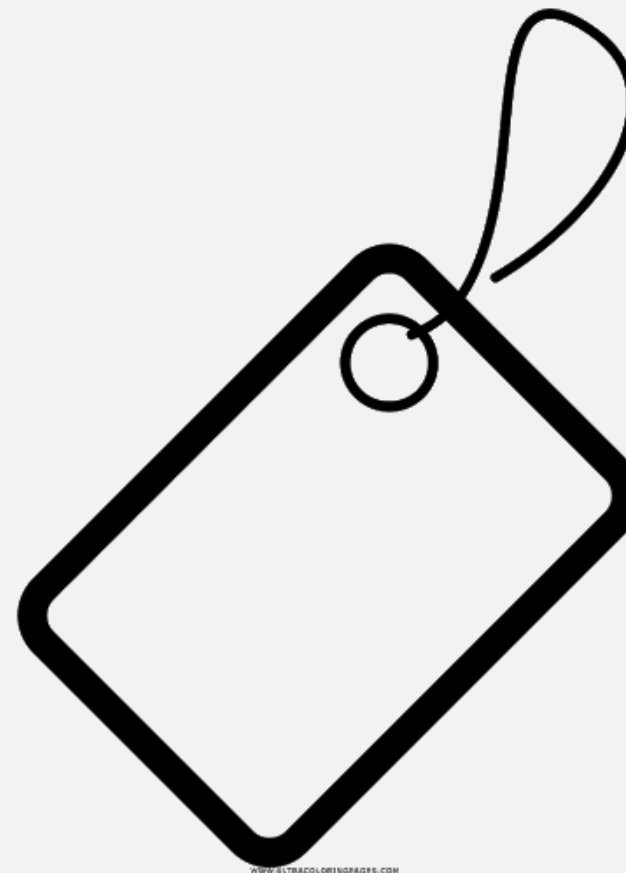
`{.js}`
JavaScript



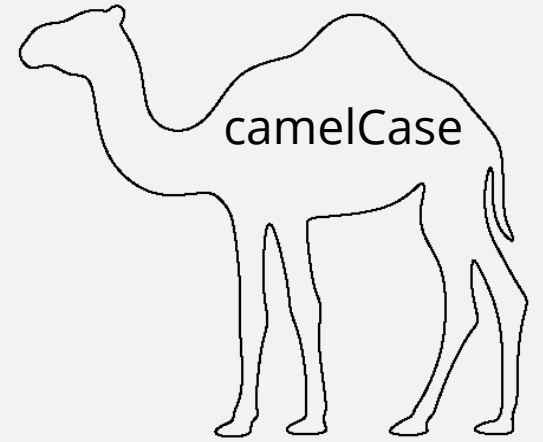
UNIÓ EUROPEA
Fons Europeu de
Desenvolupament Regional

01

Identificadors



Identificadores



Un **identificador** és un nom que se li dóna als diferents components d'un programa

- classes
- variables
- constants
- objectes
- funcions

per poder referir-nos a ells..

Els identificadors segueixen certes normes:

- **No** poden començar per un dígit.
- **Poden** estar formats per **lletres, díigits, _ i \$**.
- **No** poden contenir cap dels caràcters especials. (Diapositiva 4)
- **No pot ser una paraula reservada de JavaScript.**

JavaScript diferencia majúscules i minúscules, per tant, **casa** i **Casa** són identificadors diferents.



Caràcters especials en JavaScript

Els Identificadors en JavaScript **no poden contenir** cap dels caràcters especials següents:

+ - * / = % & # ! ? ^ " ' ~ \ | < > () [] { } : ; . ,

Exemple d'identificadors NO vàlids:



#unitats:

Invàlid ! Conté el caràcter especial #

"Cognoms":

Invàlid ! No pot contenir cometes

Paco's:

Invàlid ! Conté el caràcter '

02

Variables



Variables en JavaScript

Les **variables** són uns elements de programació encarregades de contenir les dades que s'utilitzen en el programa.



JavaScript té dos tipus de declaracions de variables:

Amb la paraula clau **var.**

Aquesta sintaxi es pot fer servir per declarar variables locals i globals, depenent del context d'execució.

Amb la paraula clau **let.**

Aquesta sintaxi es pot utilitzar per declarar una variable local amb àmbit de bloc.

És la forma moderna de declarar variables

```
var or let nom_variable = contingut;
```



```
var edat = 15;
```

```
let nom = "Pepito";
```

1

Exemples

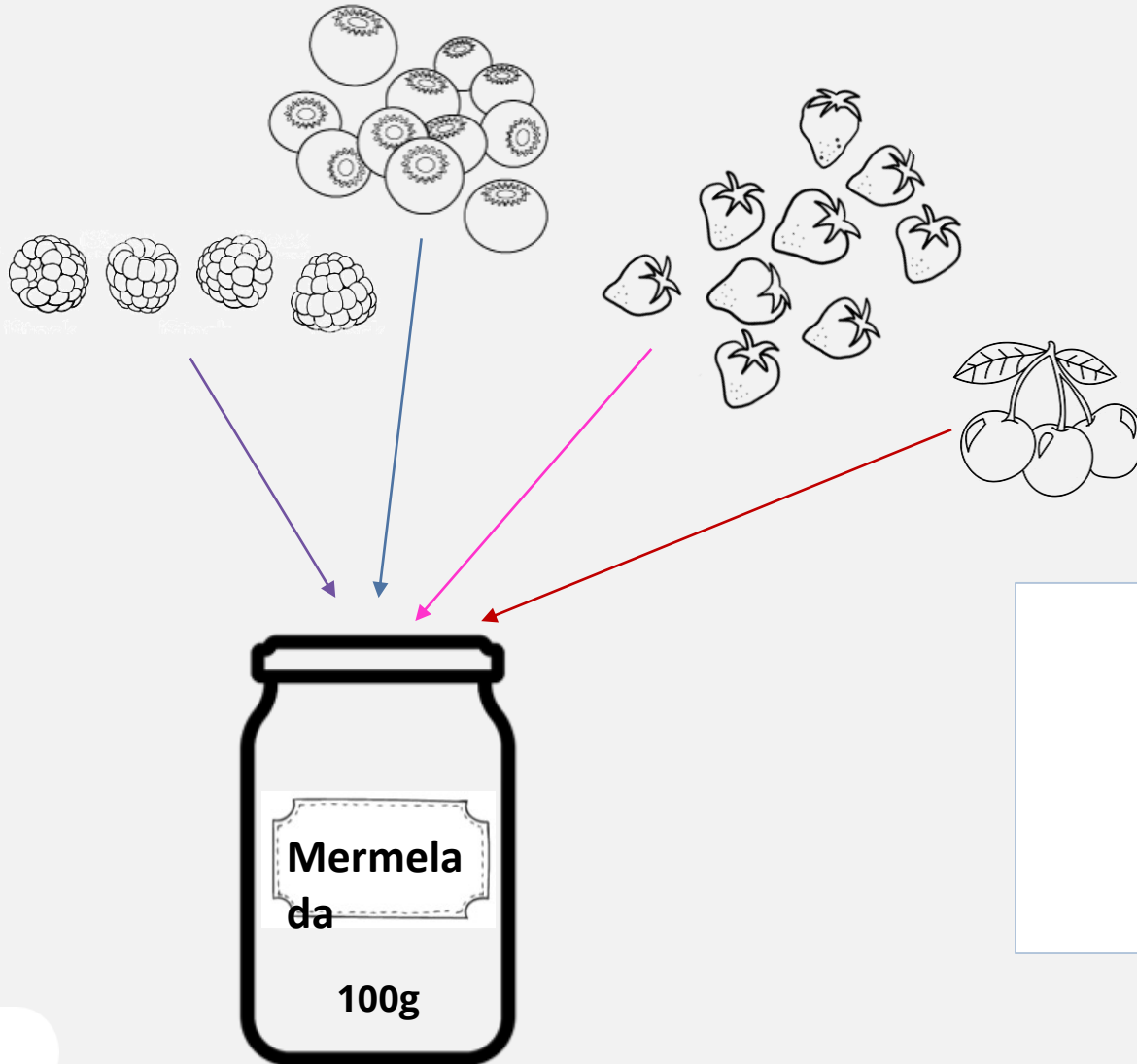
2

```
var or let nom_variable;  
... un altre codi ...  
nom_variable = contingut;
```

```
var edat;  
edat = 15;
```

```
let nom;  
nom = "Pepito";
```

Variables en JavaScript



El **contingut** d'una **variable**
pot canviar
(i generalment ho fa)
al llarg de el programa.



Tipus de variable en JavaScript



Quan es declara una variable, si no se li ha assignat cap valor inicial, contindrà el valor **null**.
Posteriorment., quan se li assigna un valor, la variable passarà a tenir assignat un determinat tipus de dades.

Quan parlem de **tipus** ens estem referint
al que pot guardar la variable ...



Tipus de variable en JavaScript

JavaScript té **tipus dinàmics**.

Això vol dir que la mateixa variable es pot utilitzar per contenir diferents tipus de dades:

- Les que guarden dades de tipus **bàsic o primitiu**
 - Nombres
 - Dades lògics o booleans
 - Strings
 - Undefined
- Les **no primitives** o referències a objectes
 - Per exemple Arrays i Classes

L'operador **typeof** pot retornar el tipus d'una variable amb dades primitius.

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

<script>
  let nom = "Pepito";
  document.getElementById("demo").innerHTML= typeof(nom);
</script>
```

Constants en JavaScript

Una **constant** és un contenidor per guardar en memòria certs valors que no es vol que canviïn durant la seva execució.

Una constant en JavaScript es declara de manera similar a una variable, precedint a aquesta, la paraula **const**.

Exemples:

```
const PI = 3.141591d;  
const DIES = 7;  
const ORGANITZACIO = "IT Academy";
```

La directiva **use strict**

Per evitar el problema de la sobrecàrrega de variables globals, a partir de la versió 5 d'ECMAScript existex el mode estricte, que no permet utilitzar variables no declarades anteriorment.

Cal afegir al nostre codi la següent línia:

```
<script>  
  "use strict";  
  // codi JavaScript  
</script>
```

Conversion de Tipus

Hi ha casos on necessitem convertir de manera explícita un valor al tipus esperat. Per això, JavaScript té tres funcions de conversió de tipus JavaScript té tres funcions de conversió de tipus

Función	Descripció	Exemple
String	Permet convertir un valor a text	<code>valor = String(valor);</code>
Number	Converteix un valor a numèric. Si el valor no és un número vàlid , el resultat de la conversió serà NaN . Si valor=null el resultat serà 0 true i false x es converteix en 1 i 0	<code>valor = Number(valor);</code>
Boolean	Converteix un valor a lògic Els valors que són intuïtivament buits , com 0, "", null, undefined, i NaN, es converteixen en false . Altres valors es converteixen en true .	<code>valor = Boolean("hola");</code> <code>alert(valor); // true</code>