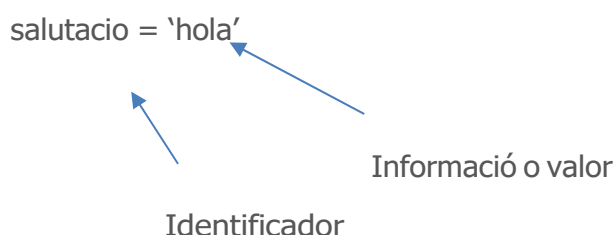


LES DIFERENTS VARIABLES I LES SEVES OPERACIONS

En aquest article explorarem què són les variables, descriurem les seves característiques i explicarem les operacions que podem realitzar-hi. Bàsicament, les variables serveixen per assignar un espai de memòria a l'ordinador per tal de preservar una informació o, millor dit, per preservar el contingut d'un objecte. A Python, qualsevol entitat és un objecte.

Així, doncs, una variable respecte al codi estarà formada pel seu identificador (nom assignat) i el valor o dada de la informació (objecte) que contingui. Exemple:



salutacio = 'hola'

Identificador

Informació o valor

A continuació, exposarem els diferents tipus d'objectes més habituals que pot guardar una variable. Són:

Cadena de caràcters (*strings*)

Aquí trobaríem les paraules, els textos, adreces de correu o pàgines web (URL) o, qualsevol altra informació que contingui una cadena de caràcters. De fet, qualsevol cadena de caràcters que faci referència a una informació i que estigui continguda entre cometes simples o dobles serà entesa per Python com una *string*.

Exemples:

```
ciutat naixement='Barcelona'
llengua='Català'
clau accés="htrn!5645@#GF[[]}**"
url='http://www.barcelonaactiva.cat'
```

Números enters (*int*)

Són aquells números positius i negatius que no tenen decimals. S'inclou també el zero.

Exemple:

```
edat=21  
errades=0
```

Números decimals (*float*)

Són aquells que tenen decimals, ja siguin positius o negatius.

Exemples:

```
puntuacio mitjana=3.50
```

Veritat o fals (*boolean*)

Aquest tipus d'informació només pot tenir el valor de veritat (*True*) o fals (*False*).

Exemple:

```
inscripcio realitzada=True  
pagament efectuat=False
```

Altra informació, com poden ser les funcions.

Exemple:

```
primera accio=obrir  
segona accio=tancar
```

Una nova variable pot contenir el valor de la informació d'una altra. Així que, per exemple, podem escriure el següent:

```
municipi actual=ciutat naixement
```

El valor que conté la variable *municipi_actual* és 'Barcelona' (una cadena de caràcters o *string*) i el *pes_minim* té un valor assignat de 3,50 (un número amb decimals o *float*).

Nota:

A partir d'ara, farem servir el nostre editor de codi i veurem resultats per pantalla. Només cal saber el següent:

`print()` – mostra per pantalla el valor que conté dins dels parèntesis.

També és important que sàpigues que, si vols afegir comentaris al teu codi per tal de fer aclariments al respecte, ho pots fer. Només cal començar la línia amb un símbol `#` o escriure el text entre `"""..."""`. Aquests comentaris no seran executats per l'interpret i ens serviran perquè els nostres companys i companyes que treballen amb nosaltres en el mateix projecte puguin seguir més fàcilment el nostre desenvolupament. Els programadors i les programadores treballen normalment en equip. És així com es poden arribar a crear grans projectes.

Una vegada hem definit el tipus d'informació o objectes amb els quals treballarem i els hi hem assignat un espai a la memòria mitjançant una variable, vegem quin tipus d'operacions permeten fer.

Enters (*int*) i números decimals (*float*)

Els números permeten fer qualsevol de les operacions matemàtiques dels seus grups. Les més bàsiques són la suma (+), la resta (-), la multiplicació (*), la divisió (/), etc.

Exemples:

```
"""calquem la nota mitjana dels alumnes"""
joan=8.3
carla=9.2
adria=7.5
dolors=6.9

suma notes=joan+carla+adria+dolors
n alumnes=4
mitjana=suma notes/ n alumnes
print(mitjana)
```

Important:

Les operacions segueixen l'ordre de jerarquia convencional.

[https://ca.wikibooks.org/wiki/Matem%C3%A0tiques_\(nivell_ESO\)/Jerarquia_de_les_operacions](https://ca.wikibooks.org/wiki/Matem%C3%A0tiques_(nivell_ESO)/Jerarquia_de_les_operacions)

A Python, els decimals s'escriuen sempre amb un punt i no amb coma.

Sobre les divisions, cal destacar que, si dividim dos números enters, el resultat sempre serà reconegut per Python com un número del tipus *float*, tot i que la divisió doni un resultat enter. Exemple:

```
alumnat=10
grups=2
alumnat per grup=alumnat/grups
print (alumnat per grup)
```

El resultat, tot i ser un 5 rodó, Python el reconeix com a 5.0.

Cadena de caràcters:

Les cadenes de caràcters també permeten la suma i la multiplicació. Comprovem-ho amb un exemple:

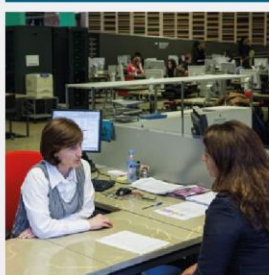
```
#sumen dues paraules i després les multipliquem per 2
p1='hola'
p2='barcelona'
salutació=p1+" "+p2
print (salutació)
print((salutació)*2)
```

Veuràs que un espai també pot ser representat com a caràcter. A la funció *print()*, es pot imprimir una seqüència d'*strings* separada per comes. Aquestes comes representaran un espai a la pantalla.

Com acabes de veure, la funció *print()* accepta operacions dins dels seus parèntesis. Això vol dir que, primer, resol l'operació i, després, n'imprimeix el resultat.

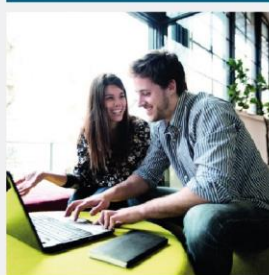
Amb això ja pots començar a escriure codi. Queda't amb nosaltres i aprendràs encara molt més.

Descobreix tot el que Barcelona Activa pot fer per a tu



Acompanyament durant tot el procés de recerca de feina

barcelonactiva.cat/treball



Suport per posar en marxa la teva idea de negoci

barcelonactiva.cat/emprenedoria



Serveis a les empreses i iniciatives socioempresarials

barcelonactiva.cat/empreses

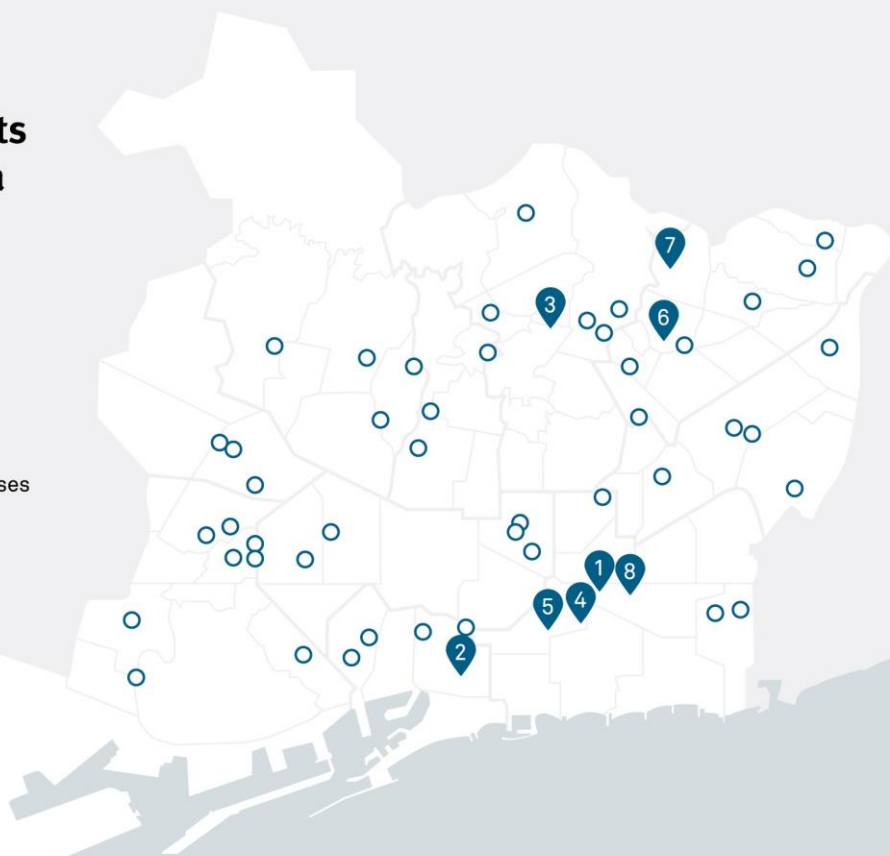


Formació tecnològica i gratuïta per a la ciutadania

barcelonactiva.cat/cibernarium

Xarxa d'equipaments de Barcelona Activa

- 1 Seu Central Barcelona Activa
Porta 22
Centre per a la Iniciativa
Emprenedora Glòries
Incubadora Glòries
- 2 Convent de Sant Agustí
- 3 Ca n'Andalet
- 4 Oficina d'Atenció a les Empreses
Cibernàrium
Incubadora MediaTIC
- 5 Incubadora Almogàvers
- 6 Parc Tecnològic
- 7 Nou Barris Activa
- 8 innoBA
- Punts d'atenció a la ciutat



© Barcelona Activa
Darrera actualització 2022

Cofinançat per:



UNIÓ EUROPEA
Fons Europeu de Desenvolupament Regional

Segueix-nos a les xarxes socials:



barcelonactiva.cat/cibernarium



[barcelonactiva](https://facebook.com/barcelonactiva)



[barcelonactiva](https://twitter.com/barcelonactiva)



company/barcelona-activa