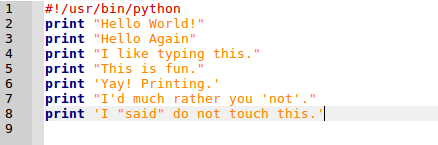
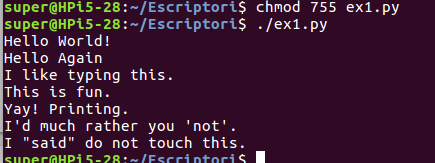
**Sesión 1**

**Ex1 - Strings**

Vamos al «Geany» y escribimos el texto:



En el terminal tan sólo debemos dirigirnos al directorio «cd Escriptori» y ejecutar lo que hay escrito en la captura de pantalla



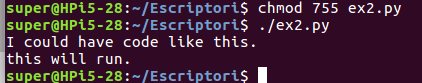
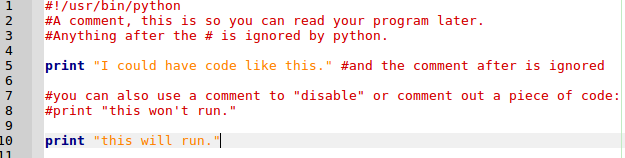
**Els strings es poden delimitar de dues maneres. Explicar quines són i la**

**combinació que se'n fa a l'exercici.**

- Los string se pueden delimitar mediante comillas y comillas simples. " y ‘

- Se combina de formas diferentes. Encontramos que si a principio y a final hay unas comillas "" se printa toda la cadena de carácteres que haya ahí dentro. Ocurre lo mismo si a principio y a final hay las simples, pues imprimirá todo el contenido que haya entre ‘ y ‘

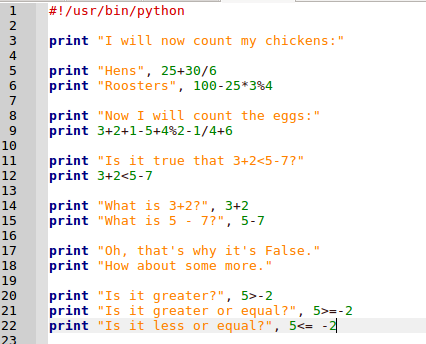
**Ex2- Comentarios**

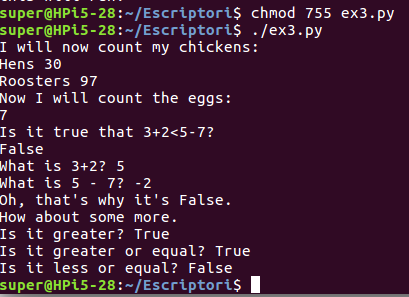


**Explicar com s'escriuen comentaris en Python. Si trobem la instrucció. Quin efecte hi tindrà el caracter #?**

- Los comentarios en Python se hacen mediante #. En el caso del ejemplo, se está mostrando por pantalla Hi # there, por lo que el símbolo # tiene un efecto nulo

**Ex3- Operaciones aritméticas y comparaciones**

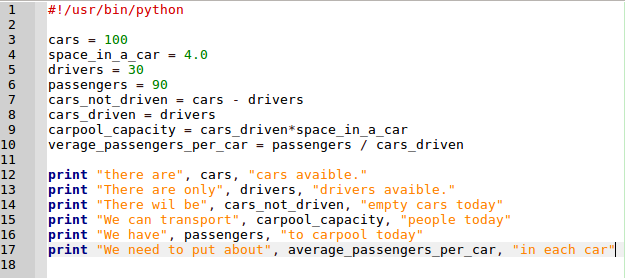




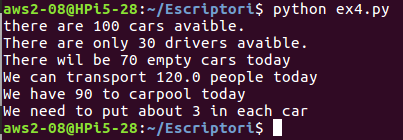
**Explicar perquè s'obtenen els resultats True i False de l'exercici.**

- Porque el signo < devuelve un boolean, pues sólo pueden ser true o false, ya que se están comparando operaciones

**Ex4- Variables y nombres. Normas de escritura**

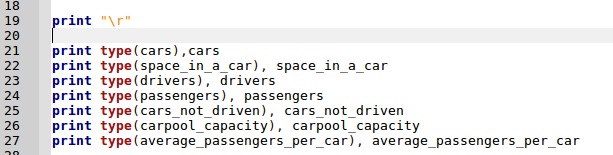
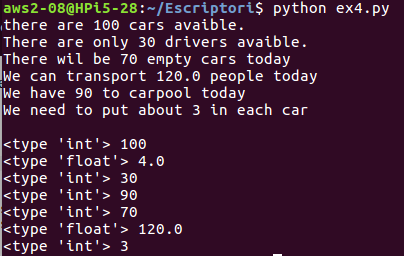


Resultado:



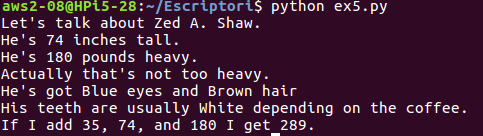
**Utilitzar la funció type i esbrinar el tipus que conté cada variable al final de**

**l'execució de l'script.**



**Ex5- Variables y escritura con formato**

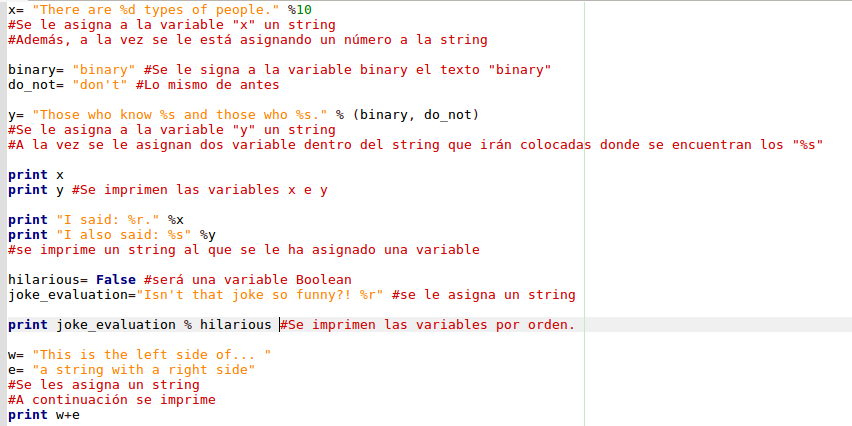


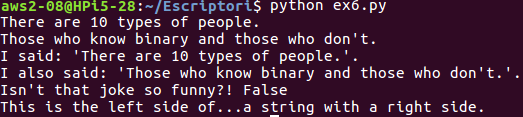


**Explicar quina és la funció de %, %s i %d en la sentència print**

- La función «%» te imprime la variable, mientras que «%s» imprime una cadena de texto. El signo «%d» imprime un número entero. Funciona como C.

**Ex6- Combinación de strings y formatos**





**Posar comentaris al codi, explicant què fa cada sentència.**

El format %r dóna el valor “raw”, cru, sense modificar, dels objectes o variables. Moltes

vegades el resultat de %d i %r és el mateix, però no sempre. En general %r va bé per veure

les dades quan es depuren els programes i els altres formats per mostrar-les a l'usuari.

Exemple on es veu clarament la diferència:

>>> import datetime

>>> x = datetime.date.today()

>>> print x

2013-01-11

>>>

>>>

>>> print "Today's date is %s ..." % x

Today's date is 2013-01-11 ...

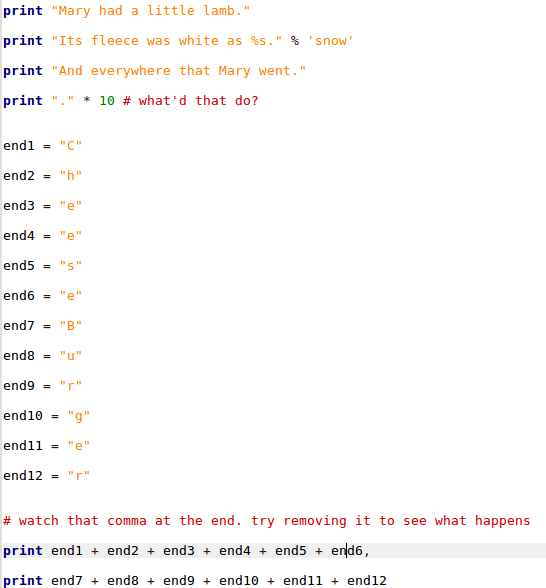
>>>

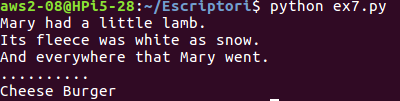
>>> print "Today's date is %r ..." % x

Today's date is datetime.date(2013, 1, 11) ...

>>>

**Ex7 – Impresión de Strings. Concatenación**



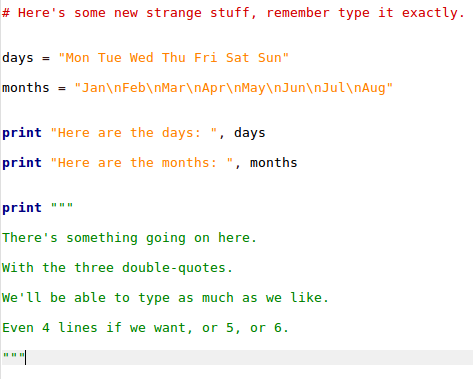


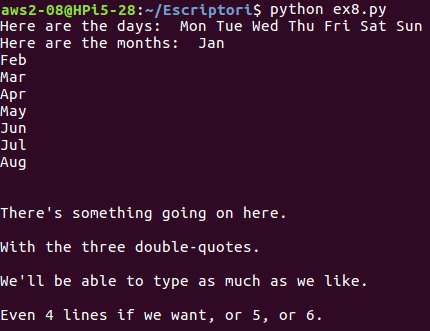
**Explicar la sentència de la línia 4 i què passa si es treu la coma final de la línia**

**20. Quina utilitat pot tenir?**

- La sentencia de la línia 4 hace que se imprima el punto 10 veces gracias ala multiplicación. Si se le quita la coma, los dos prints se hacen en líneas diferentes y eso sirve para poder tener prints diferentes si se desea en una misma línia o no.

**Ex8 – Más impresión de Strings**

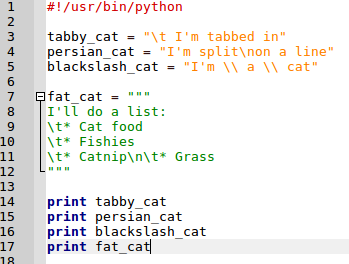


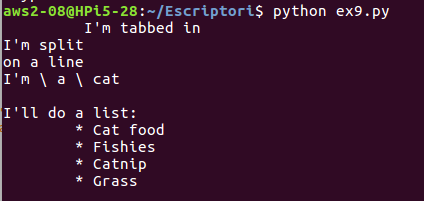


**Executar-lo. Explicar quin és l'efecte de \n i de les tres cometes dobles.**

- Cuando se usa \n se realiza un salto de línia, peró cuando se usan las tres comillas dosbles (""") podemos escribir un texto en líneas diferentes.

**Ex9- Carácteres escapados con la contrabarra**





**Executar-lo. Explicar quin és l'efecte de la contrabarra en cadascun dels strings on apareix.**

- La \t tabula el texto

- La \ entre medio de una string corta la línia

- La \\ doble, hace que no corte la línia y que además, aparezca el signo allí donde la hemos colocado

Además, \n también corta la línia, es como hacer un «intro»

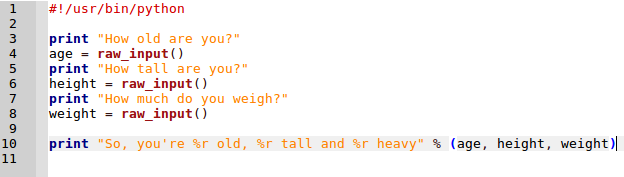
**Llistar les “seqüències d'escape” que hi ha a Python. Buscar a la documentació**

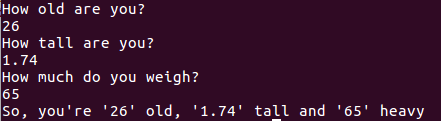
**oficial: docs.python.org.**

- No ha sigut possible poder llistar les seqüencies dins de la pàgina, però he trobat una web on estava tot ben explicat: <https://123programando.wordpress.com/tag/secuencias-de-escape/>

|  |  |  |
| --- | --- | --- |
| **Secuencia de Escape** | **En pantalla…** | **O lo que es lo mismo…** |
| \\ | \ | Una única barra invertida |
| \’ | ‘ | Escape de comilla simple en una cadena |
| \” | “ | Escape de comilla doble en una cadena |
| \a | BELL | ASCII bell |
| \b | BS | ASCII retroceso |
| \f | FF | ASCII avance de página |
| \n | LF | ASCII linefeed o salto de línea |
| \N{name} | Sólo Unicode | Carácter name en la base de datos Unicode |
| \r | CR | ASCII retorno de carro |
| \t | TAB | ASCII tabulación horizontal |
| \uxxxx | Sólo Unicode | Carácter con 16-bits hexadecimal y valor xxxx |
| \Uxxxxxxxx | Sólo Unicode | Carácter con 32-bits hexadecimal y valor xxxxxxxx |
| \v | \v | ASCII tabulación vertical (VT) |
| \ooo | oo | Carácter con valor octal |
| \xhh | hh | Carácter con valor hexadecimal |

**Ex10- Leer datos del usuario I**



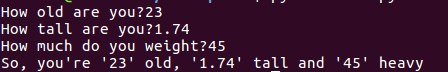


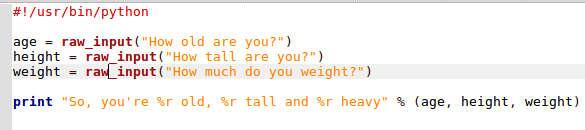
**Explicar la utilitat de les comes finals en les sentències print.**

- En el caso de usar no usar comas al final del print, hace que la respuesta del teclado se haga en otra línia, mientras que el uso de las comas al final del print, hace que la respuesta entrada por teclado se haga en la misma línia que el mensaje del print

**Si el text escrit al final no és satisfactori, provar de canviar els formats.**

**Ex11- Leer datos del usuario II**





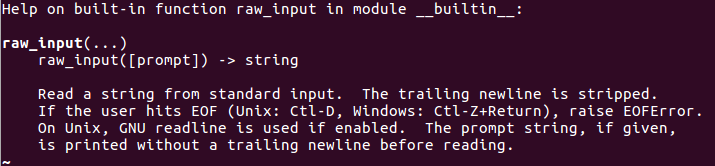
**Comparar el resultat amb l'obtingut en l'activitat anterior.**

- Realmente es lo mismo, però de forma abreviada

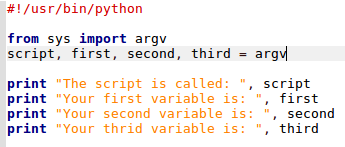
**Buscar i enganxar les característiques de la funció raw\_input() documentació oficial.**

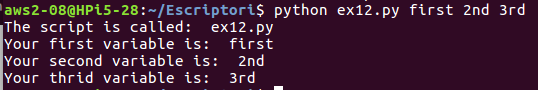
**Fer-ho també utilitzant la comanda pydoc a la consola o terminal:**

*$ pydoc raw\_input*



**Ex12- Paso de parámetros a un script por línia de comandos**



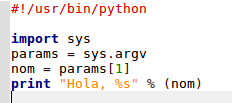


Llamas al ejercicio y le pasas como parámetro «first 2nd y 3rd. Argv es un sistema de transferencia de datos por parámetro

**Provar i explicar què passa si es passen menys valors dels esperats.**

- Da error y te remite a la línia de argumentos, en mi caso la línia 4, donde el «argv»

**Escriure aquest altre script ex13.py. En aquest cas s'importa tota la llibreria sys i s'accedeix a argv sabent que és una llista. La posició 0 conté el nom de l'script i la 1, 2 etc els corresponents paràmetres:**





**Explicar les diferències que hi ha entre l'script ex12 i l'ex13.**

- En el ex12, puedes llamar al script y printas el nombre del script seleccionado. Le has pasado por línia de parámetros en el orden en el que quieres que muestre los datos introducidos

- La manera de importar el agrv y la toma de datos varía y en este caso imprime el primer dato introducido, ya que le has dicho: params[1] (primer dato introducido).