



# ragazze DIGITALI

IDEE PER UN FUTURO SMART

# Cosa faremo oggi

Ti trovi in una terra piena di draghi.  
Di fronte a te ci sono due grotte  
In una grotta si trova un drago simpatico e socievole che  
condividerà con te il suo tesoro.  
Nell'altra grotta c'è invece un drago affamato e vorace.  
Dentro quale grotta vuoi entrare? (1 o 2)

1

Entri nella caverna 1  
E' buia e spaventosa  
Un enorme drago compare all'improvviso davanti a te! Apre  
le sue fauci e... ti inghiottisce in un batter d'occhio!  
Vuoi giocare ancora?  
No

## Serviranno le funzioni di alcuni moduli

Cosa dobbiamo fare per poter richiamare funzioni di un modulo/libreria?

random

Utilizzeremo la funzione randint() che abbiamo già visto

time

Utilizzeremo la funzione sleep(forSeconds)

# Il regno dei draghi - Codice

```
import random  
import time
```

# Outline

- 1 Funzioni
- 2 Variabili locali e variabili globali
- 3 Operatori booleani
- 4 Costrutto while
- 5 Il regno dei draghi

- Abbiamo già visto e usato alcune funzioni, ad esempio: `print('ciao')`, `input()`, `randint(1, 20)`, ecc..
- Possiamo anche crearne noi!
- Con le funzioni possiamo incapsulare del codice e riutilizzarlo più volte nel nostro programma.

## Una funzione

- Deve essere dichiarata **prima** di essere chiamata
- Può avere dei parametri
- Restituisce un risultato

## Come si dichiara una funzione

- Utilizziamo la keyword **def**
- seguita dal **nome** della funzione
- Seguita dai **parametri** racchiusi tra parentesi (o le sole parentesi aperta e chiusa nel caso non ci siano parametri)
- Seguiti da :

def keyword  
↓  
def displayIntro():  
↑                    ↑  
Function name      Colon

```
def displayIntro():  
    print('''Ti trovi  
           in una terra  
           piena di  
           draghi...''')  
print()
```

! notiamo il `print('' '')`, stringa multilinea

## Alcuni consigli..

- Ciò che abbiamo detto per i nomi delle variabili vale anche per i nomi delle funzioni
- Meglio se le nostre funzioni sono corte, con poco codice
- Con 1 funzione facciamo 1 cosa soltanto!
- Evitiamo di usare più di 1 o 2 parametri in una funzione  
Se abbiamo bisogno di più parametri dividiamo quello che vogliamo fare in più funzioni



## Funzioni con parametri

- Possiamo "dare in pasto" ad una funzione dei dati, delle informazioni, che chiamiamo **parametri**
- Queste informazioni ci serviranno per elaborare il risultato della funzione

```
def isPositive(number):  
    if number > 0 :  
        return True  
    else:  
        return False
```

# 0 ancora meglio..

```
def isPositive(number):  
    return number > 0
```

```
# Usiamo la funzione, ad esempio, in questo modo  
print(isPositive(5))      # True  
print(isPositive(-5))    # False
```

## Funzioni con parametri

- Possiamo "dare in pasto" ad una funzione dei dati, delle informazioni, che chiamiamo **parametri**
- Queste informazioni ci serviranno per elaborare il risultato della funzione
- Il risultato verrà restituito utilizzando la keyword **return**

```
def sum(firstNumber, secondNumber):  
    return firstNumber + secondNumber
```

```
# Usiamo la funzione, ad esempio, in questo modo  
result = sum(5, 4)      # result avra' valore 9  
result = sum(-10, 5)   # result avra' valore -5
```

# Outline

- 1 Funzioni
- 2 Variabili locali e variabili globali
- 3 Operatori booleani
- 4 Costrutto while
- 5 Il regno dei draghi

# Variabili locali e variabili globali

## Variabili locali

Chiamiamo **Variabile locale** qualunque variabile dichiarata all'interno di una funzione, questa esiste solo all'interno della funzione stessa. Le variabili locali vengono "dimenticate" dopo che la funzione ha raggiunto il return (e quindi ha finito le sue elaborazioni).

```
def welcomePerson():  
    person = 'Chiara'      # Variabile locale  
    print('Benvenuta ' + person)  
# Anche se non c'è il return, da qui in poi la  
# variabile non ha più effetto
```

```
welcomePerson()  
person = 'Sofia'  
print(person) # Sofia  
welcomePerson() # Benvenuta Chiara
```

# Variabili globali

## Variabili globali

Nell'esempio precedente `person = 'Sofia'` e' una variabile globale, ovvero una variabile che ha effetto in tutto il nostro programma. Modifichiamo leggermente l'esempio:

```
def welcomePerson():  
    print('Benvenuta ' + person)
```

```
person = 'Chiara'           # Variabile globale  
welcomePerson() # Benvenuta Chiara  
person = 'Sofia'  
welcomePerson() # Benvenuta Sofia
```

# Outline

- 1 Funzioni
- 2 Variabili locali e variabili globali
- 3 Operatori booleani**
- 4 Costrutto while
- 5 Il regno dei draghi

## Operatore and (e)

Quale frase è complessivamente vera e quale falsa?

- I gatti hanno i baffi E i cani hanno la coda
- I gatti hanno i baffi E i cani hanno le ali
- I gatti abbaiano E i cani hanno le ali

## Operatore and (e)

			Risultato
True	<b>and</b>	True	True
True	<b>and</b>	False	False
False	<b>and</b>	True	False
False	<b>and</b>	False	False



# Operatore or (o)

Quale frase è complessivamente vera e quale falsa?

- I gatti hanno i baffi O i cani hanno la coda
- I gatti hanno i baffi O i cani hanno le ali
- I gatti abbaiano O i cani hanno le ali

```
city = 'Cesena'  
result = 10 < 20 or city == 'Cesena'  
# result avra' valore?  
result = 10 > 20 or city == 'Cesena'  
# result avra' valore?
```

## Operatore or (o)

			Risultato
True	<b>or</b>	True	True
True	<b>or</b>	False	True
False	<b>or</b>	True	True
False	<b>or</b>	False	False

# Operatore not

```
city = 'Cesena'
result = not (10 < 20 or city == 'Cesena')
# result avra' valore?
result = not (10 > 20 or city == 'Cesena')
# result avra' valore?
```

# Operatore not

	Risultato
not True	False
not False	True

# Outline

- 1 Funzioni
- 2 Variabili locali e variabili globali
- 3 Operatori booleani
- 4 Costrutto while**
- 5 Il regno dei draghi

# Costrutto while

## while

- Costrutto simile al for, che abbiamo già visto
- Il while ripete il ciclo finché una determinata condizione rimane vera
- Qual è la differenza con il for?

```
while month == 'giugno' and year == '2019' :  
    codice del ciclo while
```

! come nel for, nell'if e nelle funzioni occhio all'indentazione del codice del ciclo!

# Outline

- 1 Funzioni
- 2 Variabili locali e variabili globali
- 3 Operatori booleani
- 4 Costrutto while
- 5 Il regno dei draghi

## Ritorniamo al gioco di oggi

Ti trovi in una terra piena di draghi.  
Di fronte a te ci sono due grotte  
In una grotta si trova un drago simpatico e socievole che  
condividerà con te il suo tesoro.  
Nell'altra grotta c'è invece un drago affamato e vorace.  
Dentro quale grotta vuoi entrare? (1 o 2)

1

Entri dentro la caverna 1

E' buia e spaventosa

Un enorme drago compare all'improvviso davanti a te! Apre  
le sue fauci e... ti inghiottisce in un batter d'occhio!

Vuoi giocare ancora?

No



# Il regno dei draghi - Parte 1

E' il vostro turno!

Scrivi una funzione, chiamandola come preferisci, che stampi a video il testo introduttivo del gioco:

'Ti trovi in una terra piena di draghi.

Di fronte a te ci sono due grotte

In una grotta si trova un drago simpatico e socievole che condividerà con te il suo tesoro.

Nell'altra grotta c'è invece un drago affamato e vorace.

Dentro quale grotta vuoi entrare? (1 o 2)'

## Il regno dei draghi - Parte 2

E' il vostro turno!

Scrivi una funzione, chiamandola come preferisci, che chieda al giocatore di inserire il numero della grotta nella quale vuole entrare (1 o 2) e restituisca/ritorni il numero stesso della grotta scelta

Aiuto:

```
answer = ''  
while answer != '1' or answer != '2' :  
    codice del ciclo while  
    "ripensa al primo giorno quando chiedevamo  
    all'utente il nome e stampavamo 'ciao  
    nome'"
```

## Il regno dei draghi - Parte 3

E' il vostro turno!

Scrivi una funzione, chiamandola come preferisci, che

- Prenda come parametro il numero della grotta scelto dal giocatore
- Stampi a video
  - 'Ti avvicini alla grotta numero ...' (numero della grotta inserito dal giocatore)
  - 'E' buia e spaventosa..'
  - 'Un enorme drago compare all'improvviso davanti a te! Apre le sue fauci e...'
- Dopo la stampa a video di ogni frase fai attendere 2 secondi al giocatore per aggiungere suspense utilizzando la funzione sleep del modulo time

```
time.sleep(2) # Interrompe l'esecuzione del  
              programma per i secondi specificati come  
              parametro
```

## Il regno dei draghi - Parte 3

### E' il vostro turno!

Nella funzione della pagina precedente, dopo le stampe e i `time.sleep(2)`

- Aggiungi una variabile `friendlyCave` dandole un valore random tra 1 e 2 (utilizzando la funzione `randint` del modulo `random` come mostrato nel codice qui sotto)
- Se la variabile `friendlyCave` ha lo stesso valore del numero della grotta scelta dal giocatore allora stampa  
'Hai vinto il tesoro!'  
altrimenti stampa  
'Il drago ti mangia in un boccone!'

```
friendlyCave = random.randint(1, 2)
```

## Il regno dei draghi - Parte 4

### E' il vostro turno!

Dopo aver inizializzato una variabile chiamata 'playAgain' con valore 'si', utilizza un ciclo while che si ripeta finche' il valore di playAgain e' la stringa 'si' e al suo interno

- Richiami la prima funzione che avete creato per stampare il testo introduttivo del gioco
- Crei una variabile locale con il valore della grotta scelta dal giocatore (richiamando la seconda funzione che avete creato)
- Controlli se il valore della grotta scelta sia lo stesso della friendlyCave (richiamando terza funzione che avete creato)
- Infine stampi a video la stringa 'Vuoi giocare ancora?' e riassegni/modifichi il valore della variabile playAgain con la nuova scelta del giocatore (usando la funzione input())