

Intelligent Signal Processing

Brief Introduction to Python

Angelo Ciaramella

Costrutto decisionale

```
if condizione:  
    # gruppo di istruzioni eseguite  
    # se la condizione è vera
```

Costrutto if-then

```
if condizione:  
    # gruppo di istruzioni eseguite  
    # se la condizione è vera  
else:  
    # gruppo di istruzioni eseguite  
    # se la condizione è falsa
```

Costrutto if-then-else



Costrutto decisionale

```
n = int(input('Inserisci un numero: '))
if n < 0: # se il numero è negativo
    n = -n # rendilo positivo
print('Il valore assoluto è', n)
```

Esempio di costrutto if-then

```
n = int(input('Inserisci un numero: '))
if n < 0: # se il numero è negativo
    print(n, 'è negativo')
else: # altrimenti (se non è negativo)
    print(n, 'è positivo')
```

Esempio di costrutto if-then-else

N.B.: i costrutti sono gestiti con le indentazioni



Costrutto decisionale

```
if condizionale1:  
    # gruppo di istruzioni eseguite  
    # se la condizionale1 è vera  
elif condizionale2:  
    # gruppo di istruzioni eseguite  
    # se la condizionale2 è vera  
elif condizionaleN:  
    # gruppo di istruzioni eseguite  
    # se la condizionaleN è vera  
else:  
    # gruppo di istruzioni eseguite  
    # se tutte le condizioni sono false
```

Esempio di costrutti annidati



Costrutto switch

```
# esempio in Python
if n == 0:
    print('zero')
elif n == 1 or n == 2:
    print('uno o due')
elif n == 3:
    print('tre')
else:
    print('numero diverso da 0, 1, 2, 3')
```

Esempio di switch

N.B.: in python non esiste lo switch



Ciclo for

■ ciclo for

- iterare su tutti gli elementi di un *iterabile* ed eseguire un determinato blocco di codice
 - liste, tuple, set, dizionari (restituiscono le chiavi), ecc.

```
>>> seq = [1, 2, 3, 4, 5]
>>> for n in seq:
...     print('Il quadrato di', n, 'è', n**2)
```

Esempio di ciclo for



Range

■ funzione `range`

- specificare uno valore iniziale o *start* (incluso), un valore finale o *stop* (escluso), e uno *step*, e che ritorna una sequenza di numeri interi

```
>>> range(5)
range(0, 5)
>>> list(range(5)) # conversione in lista
[0, 1, 2, 3, 4]
>>> list(range(5, 10))
[5, 6, 7, 8, 9]
>>> list(range(0, 10, 2))
[0, 2, 4, 6, 8]
```

Esempi di range



Range

```
>>> seq = [1, 2, 3, 4, 5]
>>> for n in seq:
...     print('Il quadrato di', n, 'è', n**2)
```

Costruire il ciclo usando range



Ciclo while

```
>>> seq = [10, 20, 30, 40, 50, 60]
>>> while len(seq) > 3:
...     print(seq.pop())
...
60
50
40
>>> seq
[10, 20, 30]
```

Esempio di ciclo while



Ciclo while

```
>>> # chiedi all'utente di inserire numeri finchè indovina
>>> n = 8
>>> while True:
...     guess = int(input('Inserisci un numero da 1 a 10:
...     '))
...     if guess == n:
...         print('Hai indovinato!')
...         break # numero indovinato, interrompi il ciclo
...     else:
...         print('Ritenta sarai più fortunato')
...     ...
```

Esempio di ciclo while



For-else

```
>>> n = 8
>>> for x in range(3):
...     guess = int(input('Inserisci un numero da 1 a 10: '))
...     if guess == n:
...         print('Hai indovinato!')
...         break # numero indovinato, interrompi il ciclo
...     else:
...         print('Tentativi finiti. Non hai indovinato')
... 
```

Else dopo il for



Comprehension

```
>>> # list comprehension che crea una lista di quadrati
>>> [x**2 for x in range(10)]
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
>>>
>>> # set comprehension che crea un set di cubi
>>> {x**3 for x in range(10)}
{0, 1, 64, 512, 8, 343, 216, 729, 27, 125}
>>>
>>> # dict comprehension che
>>> {c: c.upper() for c in 'abcde'}
{'c': 'C', 'e': 'E', 'a': 'A', 'b': 'B', 'd': 'D'}
```

Esempi di comprehension



Comprehension

```
-----  
# script  
squares = []  
for x in range(10):  
    squares.append(x**2)
```

```
-----  
Console  
-----
```

```
>>> squares
```

Ripetere le operazioni usando la comprehension



Funzioni

```
-----  
# script  
def calc_rect_area(width, height):  
    return width * height  
-----
```

Console

```
-----  
>>> calc_rect_area(3, 5)  
>>> calc_rect_area(width=3, height=5)  
>>> calc_rect_area(height=5, width=3)  
>>> calc_rect_area(3, height=5)
```

Esempio di funzione e passaggio di parametri



Funzioni

```
-----  
# script  
def say_hello(name = 'World'):  
    print('Hello {}'.format(name))  
-----
```

Console

```
-----  
>>> say_hello()  
>>> say_hello('Angelo')  
>>> say_hello(name='Python')
```

Esempio di funzioni e passaggio di parametri



Funzioni

```
-----  
# script  
def square(n):  
    return n**2  
-----
```

Console

```
-----  
  
>>> x = square(5)  
>>> square(square(5))  
>>> square(3) + square(4) == square(5)
```

Esempio di funzioni e passaggio di parametri



Funzioni

```
-----  
# script  
def midpoint(x1, y1, x2, y2):  
    xm = (x1 + x2) / 2  
    ym = (y1 + y2) / 2  
    return xm, ym  
-----
```

Console

```
-----  
>>> x, y = midpoint(2, 4, 8, 12)
```

Ritorno di più valori



Funzioni

```
-----  
# script  
lista = [1, 2, 3, 4, 5]  
def add_elem(seq, elem):  
    seq.append(elem)  
-----
```

Console

```
-----  
>>> lista  
>>> add_elem(lista, 6)  
>>> lista
```

Scope delle variabili



Funzioni

```
-----  
# script  
lista = [1, 2, 3, 4, 5]  
def add_elem(seq, elem):  
    seq.append(elem)  
    return seq  
-----
```

Console

```
-----  
>>> new_lista = add_elem(lista, 6)  
>>> new_lista
```

Ritorno di funzioni



Eccezioni

- Gestire le eccezioni
 - catturarle e gestirle tramite *try/except*

```
-----  
# script  
try:  
    n = int('five')  
except ValueError:  
    print('Invalid number!')
```

Console

```
-----  
>>> Invalid number!
```

Esempio di eccezione ValueError

