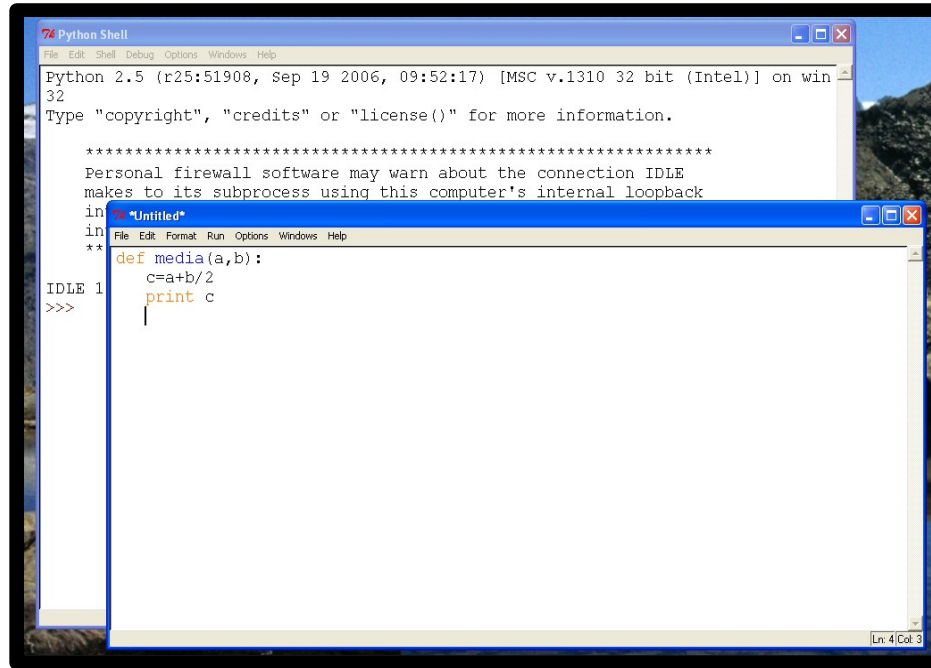


# Sommario della Seconda Lezione

- L'attivazione dell'editor di programmazione;
- Shell VS Editor;
- Esercizio guidato sul passaggio di parametri;
- Esercitazione guidata su selezioni e connettivi logici;
- Esercizi sulle selezioni e esempi di soluzione;
- Esercizi sui connettivi logici e le selezioni e esempi di soluzione;
- Semplici esercitazioni guidate sulle operazioni su stringhe
- Funzioni ... encore
- Esercitazione guidata sulla ricorsione
- Esercizi sulla ricorsione

# L'attivazione dell'editor di programmazione



**File**



**New window**

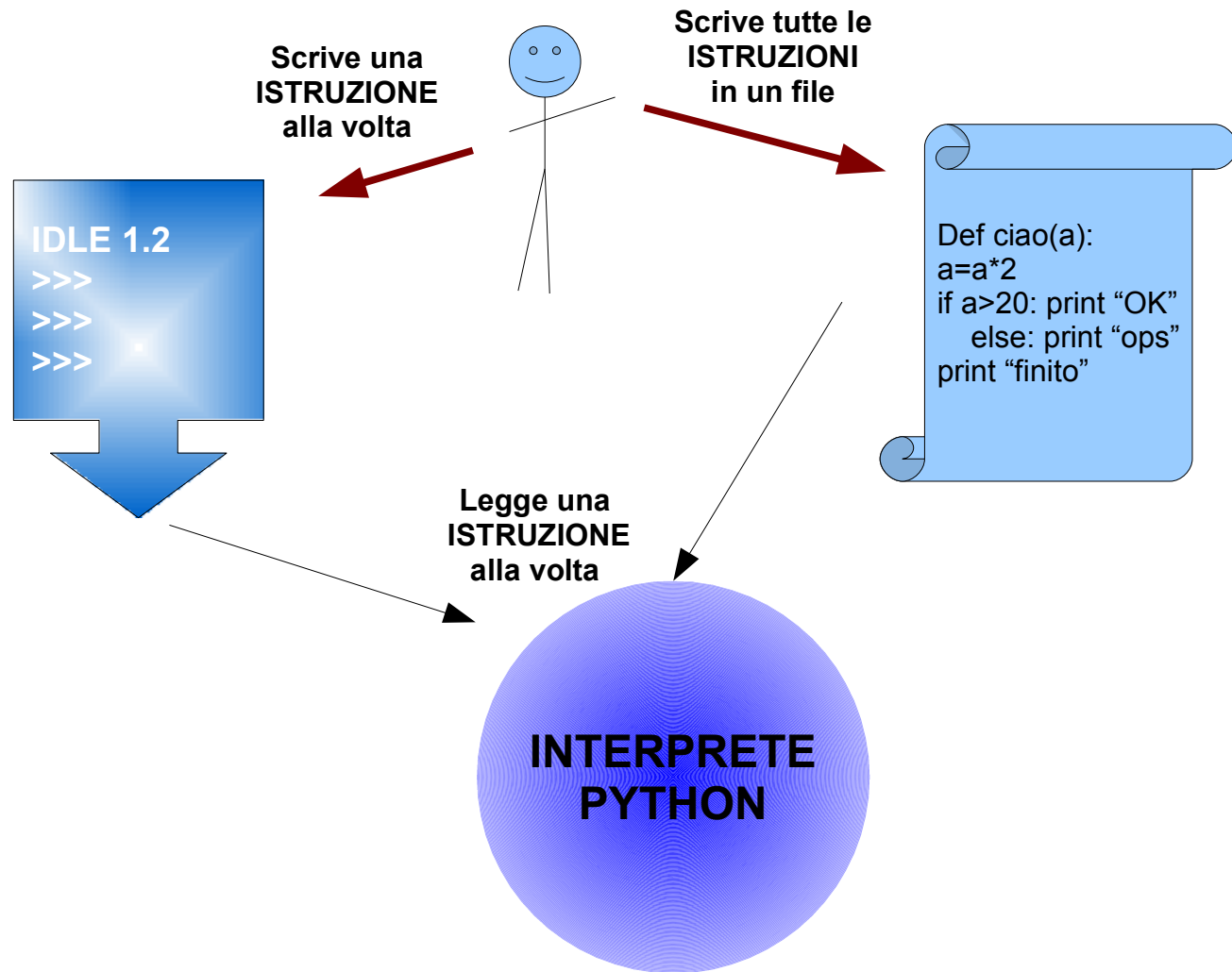
Fai interpretare il codice scritto:

**Run**



**Run Module**

# Shell VS Editor



# Esercizio guidato sul passaggio di parametri

Si definisca la seguente funzione:

```
def pippo(m):  
    print m*2
```

Si provi ora a stampare m con:

```
print m
```

m esiste al di fuori della funzione pippo ?

Il parametro “vive” solo durante l'esecuzione della funzione !!!

Si chiami ora la funzione con parametro 3 “pippo(3)”, il risultato sarà 6.

Cos'è successo dal momento che avete premuto invio ?

- l'interprete ha letto l'interfaccia della funzione;
- ha associato a m il valore 3
- ha letto l'istruzione “print m\*3”
- ha valutato m\*3
- ha attivato il comando “print” per stampare il valore valutato

# Esercitazione guidata su selezioni e connettivi logici

Si scriva la seguente istruzione dopo aver assegnato a m il valore 4:

```
m=4
if 3>m:
    print "piccolo"
```

Si assegni ora a m il valore 2 e si riprovi l'istruzione.

Si provi ora l'istruzione seguente:

```
if (m>2 and m<6):
    print m, " è compreso tra 2 e 6"
else:
    print m, " è esterno all'intervallo"
```

dopo aver posto m a 3, poi si riprovi con m=9 e m=1

Si provi anche con la condizione (m=1 or m=9):

```
if (m==3 or m==9):
    print "fuoco"
else:
    print "acqua"
```

# Esercizi sulle selezioni e esempi di soluzione

1 di 3

## *Esercizio 1:*

- a) Scrivere una funzione che prenda in input un numero intero e stampi in output “Il numero è pari” se il numero è pari, o “Il numero è dispari” altrimenti.
- b) Modifica la funzione in modo che il numero sia inserito come parametro.

## *Esercizio 2:*

- a) Scrivere una funzione che prenda in input tre voti (tre numeri interi) e stampi “ammesso all'orale” se la media è maggiore di 17.
- b) Modifica la funzione in modo che il numero sia inserito come parametro.

# Esercizi sulle selezioni e esempi di soluzione

2 di 3

## *Soluzione esercizio 1:*

### Parte A

```
def pari():  
    a = input("Inserisci un numero intero:")  
    if (a % 2 = 0):  
        print "numero pari"  
    else:  
        print "numero dispari"
```

oppure:

```
def pari():  
    a = input("Inserisci un numero intero:")  
    if (a % 2 = 0):  
        print a, "è un numero pari"  
    else:  
        print a, "è un numero dispari"
```

### Parte B

```
def pari(a):  
    if (a % 2 = 0):  
        print a, "è un numero pari"  
    else:  
        print a, "è un numero dispari"
```

# Esercizi sulle selezioni e esempi di soluzione

3 di 3

## *Soluzione esercizio 2:*

### Parte A

```
def promosso():  
    a = input("Primo voto:")  
    b = input("Primo voto:")  
    c = input("Primo voto:")  
    d=(a+b+c)/3  
    if (d >= 17): print "ammesso all'orale"
```

### Parte B

```
def promosso(a,b,c):  
    c = input("Primo voto:")  
    d=(a+b+c)/3  
    if (d >= 17): print "ammesso all'orale"
```



# Esercizi sui connettivi logici e le selezioni e esempi di soluzione

## *Esercizio 1:*

Utilizzando gli operatori di confronto '>' e '<' e l'operatore logico AND realizzare un programma che prenda in input 3 numeri interi e scriva in output il numero più grande.

## *Esercizio 2:*

Scrivere un programma che, dato in input 3 numeri interi, stampi “EUREKA” se il terzo numero è uguale alla somma dei primi 2 o al loro prodotto.

# Esercizi sui connettivi logici e le selezioni e esempi di soluzione

## *Soluzione esercizio 1:*

```
a = input('Inserisci un numero: ')
b = input('Inserisci ancora un numero: ')
c = input('Inserisci un ultimo numero: ')
if (a >= b and a >= c):
    print 'Il numero piu grande è ', a
elif ( b >= c):
    print 'Il numero piu grande è ', b
else:
    print 'Il numero piu grande è ', c)
```

## *Soluzione esercizio 2:*

```
print('Inserisci 3 numeri:')
n1 = input('1: ')
n2 = input('2: ')
n3 = input('3: ')
if ((n1+n2) == n3) or ((n1*n2) == n3):
    print 'EUREKA'
```

# Semplice esercitazione guidata sulle operazioni su stringhe

S'assegna a S1 il valore “ciao” e a S2 il valore “mondo” e si assegna a S la somma tra S1 e S2:

```
S1="ciao"  
S2="mondo"  
S=S1+" "+S2
```

Si provi a stampare il valore di L dopo ognuna delle seguenti assegnazioni “L=len(S1)”, “L=len(S2)” e “L=len(S)”.

Si stampino i primi tre caratteri di S:

```
print S[0]  
print S[1]  
print S[2]
```

Si assegna a St il valore “marzo pazzo” e si stampino i caratteri dal terzo al sesto:

```
print St[2:5]
```

dal ottavo all'undicesimo: print St[7:10]

# Funzioni ... encore

1 di 2

## PARAMETRI & VALORE RESTITUITO

Prova a scrivere la seguenti funzione:

```
def areaq(a,b):  
    d=a*b  
    return d
```

Scrivi la seguente assegnazione: `m=areaq(2,4)`

Stampa m: `print(m)`

Qual'è la funzione di “return” ?

Scrivi la seguente funzione:

```
def areacastello(L1,L2,TL):  
    a=areaq(L1,L2)+4*areaq(TL,TL)  
    return a
```

Quindi quali sono gli scopi di chi usa le funzioni ?

# Funzioni ... encore

2 di 2

## RICORSIONE

Come definite il fattoriale di N ?

$$1! = 1$$

$$N! = (N-1)! * N \text{ se } N > 1$$

In sostanza la definizione dei numeri naturali sfrutta se stessa !

Ovvero se scrivo N! in notazione funzionale:

$$\text{fatt}(1) = 1$$

$$\text{fatt}(N) = N * \text{fatt}(N-1) \text{ se } N > 1$$

Due elementi si combinano:

1. il richiamo della funzione a se stessa
2. la condizione di terminazione

# Esercitazione guidata sulla ricorsione

1 di 2

Si vuole realizzare una funzione che stampi N volte una stringa S. Sia N che S figurano come parametri della funzione. Come posso realizzarla con la ricorsione ?

```
def stampa(N,S):  
    print S  
    if N>1:  
        stampa(N-1,S)
```

Cosa succederebbe se mancasse la selezione ? Che scopo ha la selezione ?

Scrivere un programma ricorsivo che prende in input un numero ed effettui da quel numero il conto alla rovescia !

```
def rovescia(N):  
    print N  
    if (N>0):  
        rovescia (N-1)
```

# Esercitazione guidata sulla ricorsione

2 di 2

Cosa fa la seguente funzione ?

```
def m(x, y):  
    if x == 0 or y == 0:  
        return 0  
    if y == 1:  
        return x  
    return x + m(x, y-1)
```

# Esercizi sulla ricorsione

1/6

Esercizio 1:

Scrivere un programma ricorsivo che prende in input un numero e stampa in output lo stesso numero, una cifra per riga, partendo dalla prima. Dopo modifica il programma in modo che parta dall'ultima cifra.

Esempio di risultato del programma nella sua prima fase:

```
>>> scomponi(37832)
```

3

7

8

3

2

e nella sua seconda fase:

2

3

8

7

3



# Esercizi sulla ricorsione

2/6

Soluzione dell'esercizio 1:

Fase 1:

```
def scomponi(n):  
    if (n == 0):  
        return  
    print n%10  
    scomponi(n/10)
```

Fase 2:

```
def scomponi(n):  
    if (n == 0):  
        return  
    scomponi(n/10)  
    print n%10
```

# Esercizi sulla ricorsione

3/6

## Esercizio 2:

Scrivere una funzione ricorsiva che prende in input un numero e restituisce il valore True se il numero contiene uno zero tra le sue cifre, False altrimenti.

## Esercizio 3:

Scrivere una funzione che prende in input la stringa s e stabilisca se è palindroma.

PALINDROMO: parola che si legge nella stessa maniera da destra a sinistra e da sinistra a destra, esempi: ala, alla, ada, afa, osso e odo.

# Esercizi sulla ricorsione

4/6

Soluzione dell'esercizio 2:

```
def controlla(n):  
    if n == 0:  
        return False  
    if n % 10 == 0:  
        return True  
    else:  
        return controlla(n/10)
```

Soluzione dell'esercizio 3:

```
def pal(s):  
    if (s == ""):  
        return 'true'  
    if s[0] == s[len(s)-1]:  
        return pal(s[1:len(s)-1])  
    else:  
        return 'false'
```

# Esercizi sulla ricorsione

5/6

## Esercizio 4:

Scrivere una funzione ricorsiva che implementi l'elevamento a potenza, per un qualsiasi numero e un qualsiasi esponente.

Esempi:

eleva(5, 4) calcola  $5^4$

eleva(2, 3) calcola  $2^3$

# Esercizi sulla ricorsione

6/6

Soluzione dell'esercizio 4:

```
def eleva(x, y):  
    if x == 0:  
        return 0  
    if y==1:  
        return 1  
    if y == 1:  
        return x  
    return x * eleva(x, y-1)
```