

- Fonctions comme sous-programme
 - Variables (affectations)
 - Paramétrisation
- Fonctions avec paramètres

- ensemble d'instructions regroupées de manière à définir une « nouvelle instruction ».

```
def <identificateur>():  
    |  
    | <bloc>  
    |
```

Syntaxe:

- notion d'identificateur
- notion de bloc
- indentation
- Définition de la fonction : en tête de script, voire dans un fichier séparé inclus (importé)
- Appel: la fonction étant une instruction, elle est exécutée chaque fois qu'on l'appelle

$x = 750$

Syntaxe: $\langle \text{identifiant} \rangle = \langle \text{valeur} \rangle$

Une telle *affectation*

- fixe une signification à un identifiant
- (et aussi un type, deviné par python)
- permet de remplacer les valeurs numériques par un symbole plus lisible
- et de mieux gérer les modifications.

Carré de 150 pixels...

```
fd(150)
left(90)
fd(150)
left(90)
fd(150)
left(90)
fd(150)
left(90)
```

Carré de 180 pixels...

```
fd(180)
left(90)
fd(180)
left(90)
fd(180)
left(90)
fd(180)
left(90)
```

Carré de 75 pixels...

```
fd(75)
```

```
left(90)
```

Carré de “cote” pixels de côté...

```
cote = 75  
fd(cote)  
left(90)  
fd(cote)  
left(90)  
fd(cote)  
left(90)  
fd(cote)  
left(90)
```

Carré de “cote” pixels de côté...

```
cote = 75  
fd(cote)  
left(90)  
fd(cote)  
left(90)  
fd(cote)  
left(90)  
fd(cote)  
left(90)
```

Modification plus facile, lisibilité meilleure

Jusqu'où aller ?

```
cote = 75
```

```
angle = 90
```

```
fd(cote)
```

```
left(angle)
```

Jusqu'où aller ?

```
cote = 75  
angle = 90  
fd(cote)  
left(angle)  
fd(cote)  
left(angle)  
fd(cote)  
left(angle)  
fd(cote)  
left(angle)
```

Modification plus facile, lisibilité meilleure

Jusqu'où aller ?

```
cote = 75
angle = 90
fd(cote)
left(angle)
fd(cote)
left(angle)
fd(cote)
left(angle)
fd(cote)
left(angle)
```

Modification plus facile, lisibilité meilleure
le côté est un paramètre, pas l'angle (pour un carré)

Fonction « carré de 150 de côté »

```
def carre_150():  
    fd(150)  
    left(90)  
    fd(150)  
    left(90)  
    fd(150)  
    left(90)  
    fd(150)  
    left(90)
```

Fonction « carré de 150 de côté », variable “locale”

```
def carre_150():  
    lgr = 150  
    fd(lgr)  
    left(90)  
    fd(lgr)  
    left(90)  
    fd(lgr)  
    left(90)  
    fd(lgr)  
    left(90)
```

Fonction avec paramètre « carré de *lgr* de côté »

```
def carre(lgr):  
    fd(lgr)  
    left(90)  
    fd(lgr)  
    left(90)  
    fd(lgr)  
    left(90)  
    fd(lgr)  
    left(90)
```

Fonction avec paramètre « carré de *lgr* de côté »

```
def carre(lgr):  
    fd(lgr)  
    left(90)  
    fd(lgr)  
    left(90)  
    fd(lgr)  
    left(90)  
    fd(lgr)  
    left(90)
```

Appel : `carre(75)`

Fonction avec paramètre « carré de lgr de côté », d'angle agl :

```
def carre(lgr, agl):  
    fd(lgr)  
    left(agl)  
    fd(lgr)  
    left(agl)  
    fd(lgr)  
    left(agl)  
    fd(lgr)  
    left(agl)
```

Fonction avec paramètre « carré de lgr de côté », d'angle agl :

```
def carre(lgr, agl):  
    fd(lgr)  
    left(agl)  
    fd(lgr)  
    left(agl)  
    fd(lgr)  
    left(agl)  
    fd(lgr)  
    left(agl)
```

Appel : `carre(75,90)`

Fonction avec paramètre « carré de lgr de côté », d'angle agl :

```
def carre(lgr, agl):  
    fd(lgr)  
    left(agl)  
    fd(lgr)  
    left(agl)  
    fd(lgr)  
    left(agl)  
    fd(lgr)  
    left(agl)
```

Appel : `carre(75,90)`

Un cran trop loin ?

Fonction avec paramètre « carré de *lgr* de côté », d'angle *agl*:

```
def carre(lgr, agl):  
    fd(lgr)  
    left(agl)  
    fd(lgr)  
    left(agl)  
    fd(lgr)  
    left(agl)  
    fd(lgr)  
    left(agl)
```

Appel : `carre(75,90)`

Un cran trop loin ?

Que se passe-t-il si on fait l'appel `carre(75,120)` ?

Fonction avec paramètre « carré de *lgr* de côté », d'angle *agl*:

```
def carre(lgr, agl):  
    fd(lgr)  
    left(agl)  
    fd(lgr)  
    left(agl)  
    fd(lgr)  
    left(agl)  
    fd(lgr)  
    left(agl)
```

Appel : `carre(75,90)`

Un cran trop loin ?

Que se passe-t-il si on fait l'appel `carre(75,120)` ?

l'angle n'est pas un bon paramètre (pour un carré)