

Séance 2

* Fonctions :

Syntax : type nom (paramètres)

{
// corps de la fonction
return value;
}

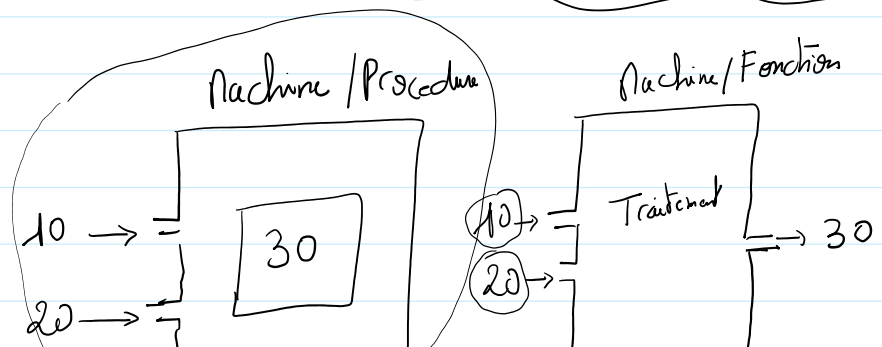
int / float / char / int[] ...

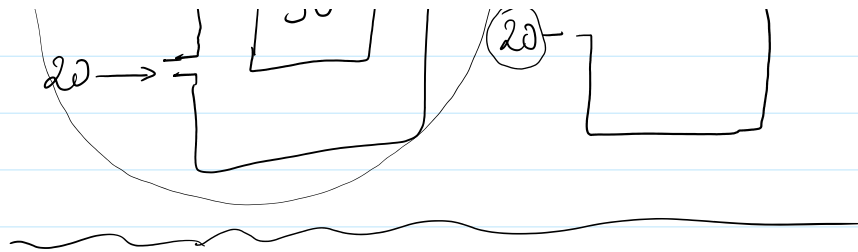
```
void somme ( int x, int y )
{
    int s;
    s = x + y;
    printf("%d", s);
}
```

} Procédure

```
int main () {
    somme(1, 2);

    return 0;
}
```



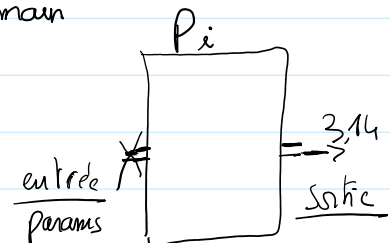


```
int somme2(int nbc1, int nbc2)
{
    int s;
    s = nbc1 + nbc2;
    return s;
}
```

```
int main() {
    int res;
    res = somme2(500, 600);
    printf("%d", somme2(5, 67));
    return 0;
}
```

Exemple: * écrivez une fonction nommée pi
qui retourne 3,14
* exploitez la dans le main

```
float pi() {
    return 3,14;
}
```



```
#include <stdio.h>
```

```
float PI(){
```

```
    return 3.14;
```

```
}
```

```
int main() {
```

```
    float rayon;
```

```
    printf("donnez le rayon de votre cercle");
```

```
    scanf("%f",&rayon);
```

```
    float surface;
```

```
    float pi = PI(); // pi en miniscule
```

```
    surface = pi * rayon * rayon;
```

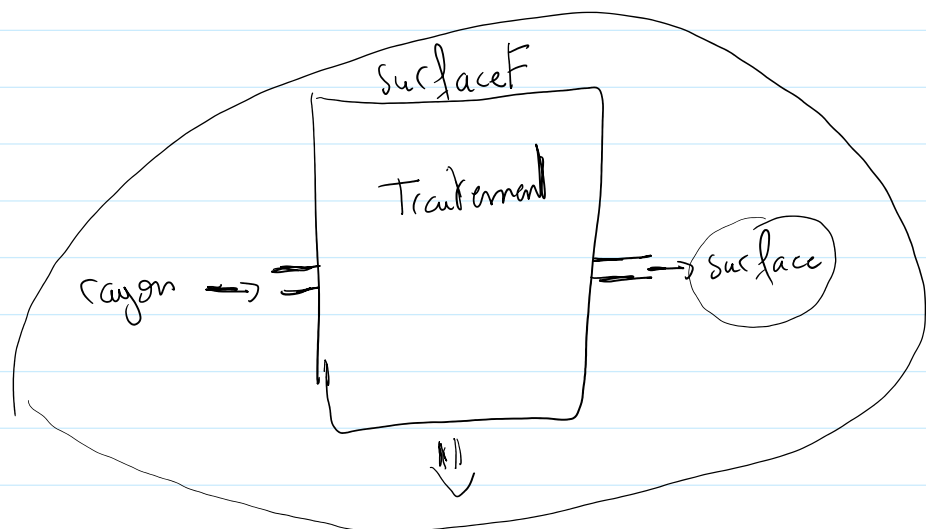
```
    //surface = rayon * rayon * PI();
```

```
    printf("la surface est %f",surface);
```

```
    return 0;
```

```
}
```

Écrivez une fct^o qui calcule
la surface d'un cercle



```
float surfaceF(float rayon) {
```

```
    float s;
```

```
    s = rayon * rayon * PI();
```

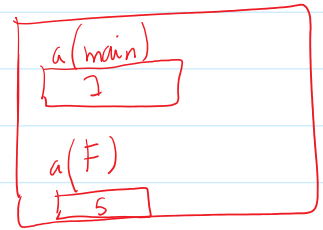
```
    return s;
```

```
}
```

int f(int a, int b) { \Rightarrow a=5 et b=6
 return f;

}
}

```
int main() {  
  int a = F(5, 6);  
  a = 7;  
}
```



}



```
int F(int x) {  
  int a = 9;  
  return 0;  
}
```

a::F

```
int main() {  
  int a = 7;   
}
```

}



```

1  #include <stdio.h>
2  float PI(){
3      return 3.14;
4  }
5  float surfaceF(float rayon){
6      float s;
7      s=rayon * rayon * PI();
8      return s;
9  }
10 int main() {
11     float rayon;
12     printf("donnez le rayon de votre cercle");
13     scanf("%f",&rayon);
14     float surface;
15     surface = surfaceF(rayon);
16     //surface = rayon * rayon * PI();
17     printf("la surface est %f",surface);
18     return 0;
19 }

```



: au moment où un return est réalisé
on quitte la fonction

```

int F(){
    return 10;
    printf("salut");
}

```

⇒ jamais exécuté

```

⇒ int main() {
    F();
    return 0;
}

```

exécution à la main:

1) bool isPair(int x) {

```
1) bool isPair(int x) {  
    if (x % 2 == 0)  
        return true;  
    else  
        return false;  
}
```

```
int main() {  
    int i;  
    for (i = 0; i <= 10; i++)  
        if (isPair(i) == true)  
            printf("%d\n", i);  
    return 0;  
}
```