

Pointeur de fonction

1. Pointeur de fonction

- En C/C++, le nom d'une fonction n'est pas considéré comme un nom de variable.
- Il est possible de définir un pointeur sur une fonction, qui n'est qu'un pointeur constant sur la première instruction de la fonction.
- On peut ainsi assigner une valeur au pointeur de fonction, le passer à une fonction, le retourner d'une fonction et, finalement le ranger dans un tableau.

1.1. Syntaxe

```
type (*identificateur) (paramètres);
```

```
void (*ptr_fonction) (int, char);
```

- La variable « ptr_fonction » est un pointeur sur une fonction qui a 2 arguments du type (int, char) et ne retourne rien.

1.2. Confusion à éviter

- Ne pas confondre pointeur sur une fonction et un pointeur classique.
- Une fonction qui retourne un pointeur du type « int ».

```
int *f();
```

- La variable « ptrf » est un pointeur sur une fonction qui n'a aucun argument et qui retourne un « int ».

```
int (*ptrf)();
```

- La variable « tab » est un tableau de 10 entiers.

```
int (*tab)[10];
```

1.3. Déréférencement d'un pointeur de fonction

```
// La fonction « affiche »  
  
void affiche();  
  
// Un pointeur de fonction, même signature que  
// la fonction « affiche »  
  
void (*ptrf) ();  
  
// Déréférencement : Initialisation de « ptrf » avec  
// l'adresse de la fonction « affiche »  
  
ptrf = &affiche;
```

1.4. Référence de fonction

- On peut déclarer aussi une référence sur une fonction.
- L'intérêt est d'éviter d'utiliser la notion de pointeur.
- L'utilisation est très limitée.