

Support de cours

Cours:

Initiation à la programmation (en C++)

Vidéo:

Prototypes (partie 1)

Concepts (extraits des sous-titres générés automatiquement) :

Exemple du cours. Facettes des fonctions. Aspect prototype. Type de la valeur de retour. Appels des fonctions. Prototype de la fonction moyenne. Parenthèses rondes. Exemple de prototype. Séquence vidéo précédente. Exemple précédent du code. Reste du programme. Cadre des fonctions. Nom de la fonction. Détails des prototypes. Calcul de la moyenne.



[vers la recherche de séquences vidéo](#)
(dans Initiation à la programmation (en C++).)



[vers la vidéo](#)

Center for Digital Education. Plus de matériel de soutien pédagogique ici :

<https://www.epfl.ch/education/educational-initiatives/cede/educational-technologies-gallery/boocs-en/>

Fonctions : prototypes

(Partie 1)

Initiation à la programmation (C++)

Vincent Lepetit, Jean-Cédric Chappelier et Jamila Sam

...

notes

résumé

0m 0s





Dans une séquence vidéo précédente, nous avons vu ce que sont les trois facettes des fonctions, nous avons aussi vu en détail, les appels des fonctions. Nous allons dans cette séquence-ci

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

0m 1s



.....

.....

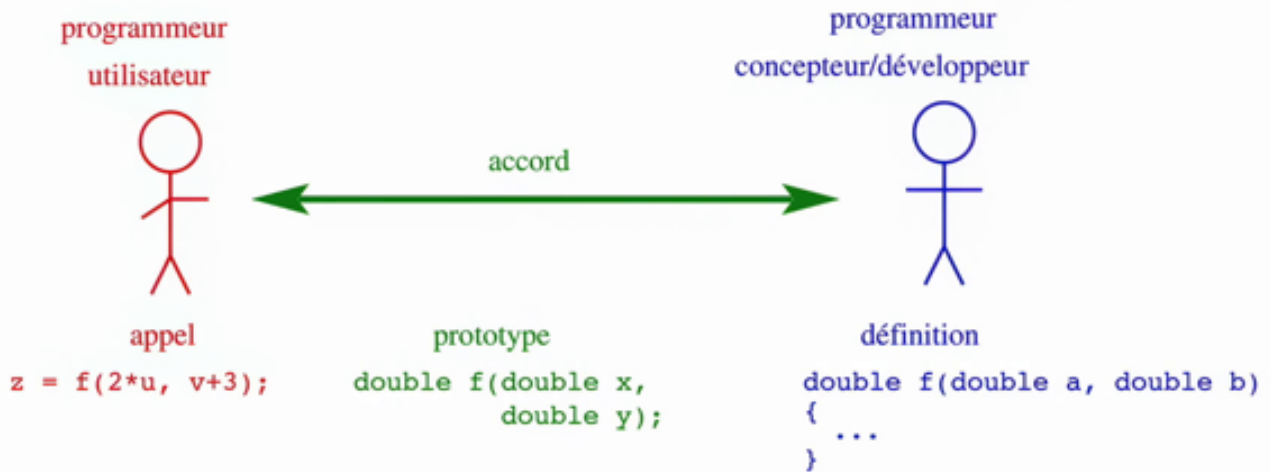
.....

.....

.....

Les « 3 facettes » d'une fonction

- ▶ **Résumé / Contrat (« prototype »)**
- ▶ Création / Construction (« définition »)
- ▶ Utilisation (« appel »)



plus particulièrement nous concentrer sur l'aspect prototype.

notes

résumé

0m 13s



Prototypage

Toute fonction doit être **annoncée** avant d'être utilisée : **prototype**

prototype = *déclaration* de la fonction, sans en définir le corps :

- ▶ nom
- ▶ paramètres
- ▶ type de (la valeur de) retour

Syntaxe : type nom (liste de paramètres);

liste de paramètres

type₁ id_param₁, ..., type_N id_param_N

Exemples de prototypes :

```
double moyenne(double x, double y);
int score (double points, double temps_jeu);
double sqrt(double x);
int demande_n
```

double moyenne (
double x,
double y) ;

entête

Dans l'exemple du cours, l'exemple avec le calcul de la moyenne, le prototype de la fonction moyenne est la ligne qui se trouve ici. Nous allons maintenant voir les détails des prototypes. En C++, tout comme les variables, les fonctions doivent être annoncées avant d'être utilisées. Dans le cadre des fonctions, c'est ce que l'on appelle le prototype. Le prototype sert donc à déclarer au reste du programme trois choses : le nom de la fonction, comment on va l'appeler, les paramètres à lui fournir pour que la fonction puisse fonctionner et enfin, le type de la valeur de retour, c'est-à-dire le type que la fonction va retourner au reste du programme. La syntaxe générale de déclaration d'un prototype de fonction est la suivante : on commence par annoncer le type de retour, ce que la fonction va renvoyer au reste du programme. Puis ensuite, on donne le nom de cette fonction. Et enfin, entre les parenthèses rondes, on va avoir la liste de tous les paramètres que la fonction doit recevoir pour fonctionner. Cette liste est éventuellement vide si la fonction n'a pas besoin de recevoir de paramètres. Comme exemple de prototype, on peut reprendre l'exemple précédent du code donné dans le transparent précédent, qui est le calcul de la moyenne. Ici, on va donc calculer la moyenne de deux nombres. Pour cela, on va dire que la fonction retourne au reste du programme, renvoie une valeur de type "double", qu'elle va s'appeler "moyenne" et qu'elle va recevoir deux paramètres. Un paramètre x de type double, et un paramètre y de type double. Et on termine la déclaration du prototype avec le point virgule ";". La partie qui précède le ";" dans un prototype, c'est ce qu'on appelle l'entête de la fonction et quand l'entête est suivie d'un ";", c'est

notes

résumé

0m 14s



Prototypage

Toute fonction doit être *annoncée* avant d'être utilisée : **prototype**

prototype = *déclaration* de la fonction, sans en définir le corps :

- ▶ nom
- ▶ paramètres
- ▶ type de (la valeur de) retour

Syntaxe : type (nom) (type₁ id_param₁, ..., type_N id_param_N) ;

liste de paramètres

Exemples de prototypes :

```
double moyenne(double x, double y);
int score (double points, double temps_jeu);
double sqrt(double x);
int demande_n
```

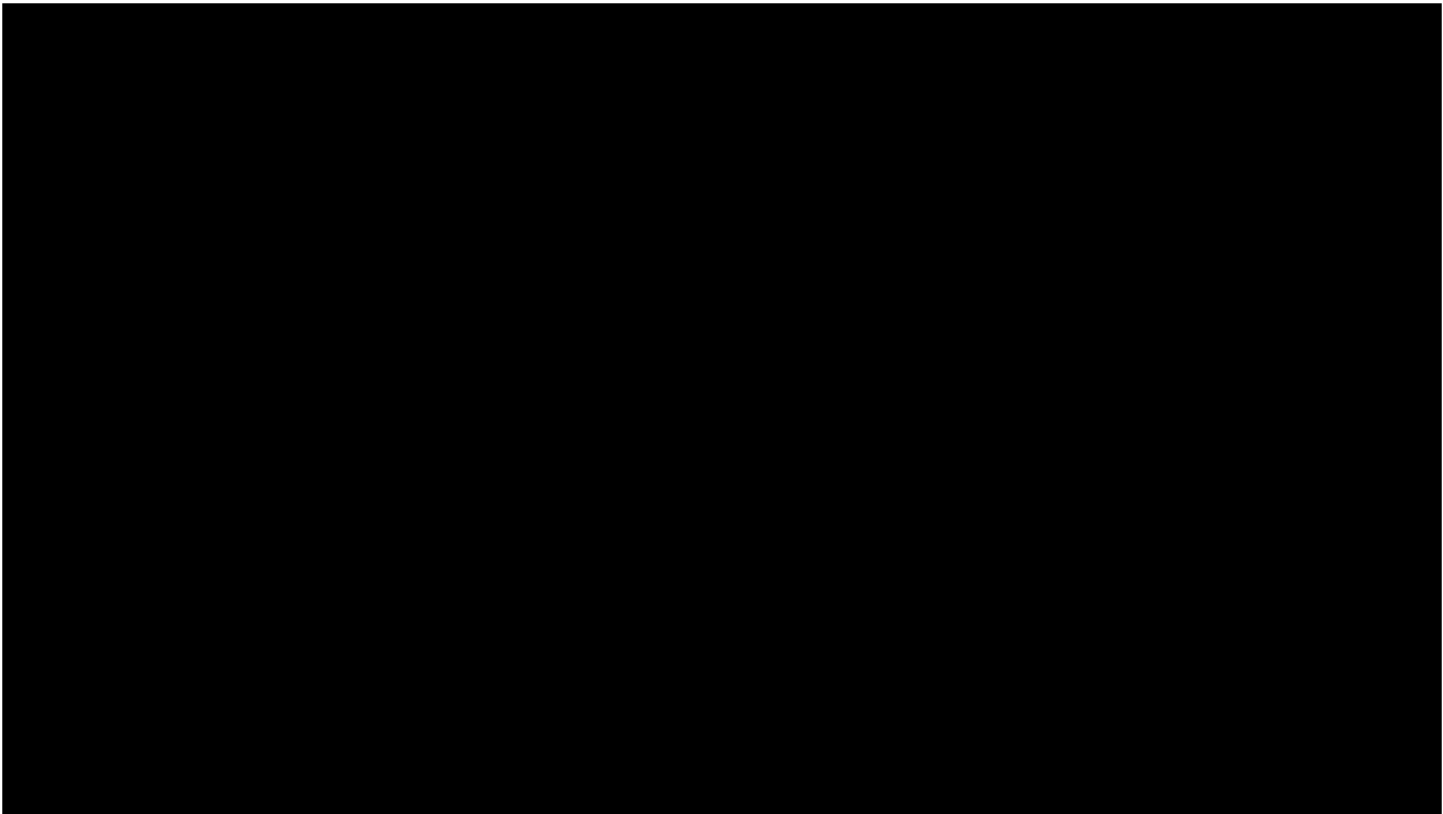
double moyenne (
double x,
double y) ;

entête

ce que l'on appelle le prototype. La liste des paramètres peut éventuellement être vide. Si par exemple, j'ai une fonction qui doit demander un entier à l'utilisateur, on pourrait par exemple lui dire qu'elle doit retourner à un entier, qu'elle doit demander un nombre.

notes

résumé



Et pour demander un nombre, elle n'a pas besoin de recevoir quoi que ce soit pour effectuer cette tâche. A ce moment-là, le prototype sera le suivant, avec ici deux parenthèses, sans liste de paramètres, deux parenthèses simplement collées. deux parenthèses simplement collées.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

2m 25s



.....

.....

.....

.....

.....