XKNATIS

Fonctions

Laurence
XKNATIS

XDNATIS

Compétence demandée :
Utiliser des fonctions dans son code

XWIKI NATIS

1. Définition
2. Construction d’une fonction
3. Traduction

XWIKI NATIS

DEFINITION

XDNATIS

Qu‘est-ce qu‘un programme ?

XWIKI NATIS

Est-ce qu‘un bloc d‘instructions est un programme ?

XWIKI NATIS

A quoi servent les fonctions en algorithmique ?

XNATIS

RAPPEL : Les fonctions sont des blocs d'instructions associés à un nom.

Ce sont des sous-programmes internes à l'application

XWIKI NATIS

Comme tous programmes, ils peuvent prendre des données en entrée, les manipuler et fournir un résultat (habituellement une donnée déduite)

XNATIS

Lorsqu'on définit une fonction, on met des variables sur lesquelles les instructions vont agir : ce sont les données en entrée, appelés aussi paramètres ou arguments

XWIKI NATIS

Les fonctions (également appelées « procédures » dans certains langages de programmation et « méthodes » dans la plupart des langages de programmation orientés objet) sont un ensemble d'instructions regroupées pour obtenir un résultat spécifique.

XWIKI NATIS

CONSTRUCTION

XEBIA NATIS

Pour construire une fonction, il faut se poser 3 questions :

- Quel est le type de résultat attendu ?
- De quoi a-t-on besoin ? (choses que l'on utilisera dans notre algorithme comme les ingrédients)
- Quel est l'algorithme pour y parvenir ?

XWIKI NATIS

Pour construire une fonction, il faut donc se poser 3 questions :

- Type de retour/résultat ?
- Paramètres ou arguments ?
- Algorithme ?

XWIKI NATIS

Pour construire une fonction, il faut donc se poser 3 questions :

- Type de retour/résultat ?
- Paramètres ou arguments ?
- Algorithme ?

XEBIA NATIS

Exemple de besoin	Type de retour
Addition de 2 nombres	nombre
Addition tous les nombres d'un tableau	nombre
Vérifier si une personne est majeur grâce à son âge	???
Appliquer une réduction de 80% à un prix	???
Ajouter un produit a un panier	???
Supprimer un produit d'un panier	???

XEBIA NATIS

Exemple de besoin	Type de retour
Addition de 2 nombres	nombre
Addition tous les nombres d'un tableau	nombre
Vérifier si une personne est majeur grâce à son âge	boolean
Appliquer une réduction de 80% à un prix	???
Ajouter un produit a un panier	???
Supprimer un produit d'un panier	???

XEBIA NATIS

Exemple de besoin	Type de retour
Addition de 2 nombres	nombre
Addition tous les nombres d'un tableau	nombre
Vérifier si une personne est majeur grâce à son âge	boolean
Appliquer une réduction de 80% à un prix	nombre
Ajouter un produit a un panier	???
Supprimer un produit d'un panier	???

XEBIA NATIS

Exemple de besoin	Type de retour
Addition de 2 nombres	nombre
Addition tous les nombres d'un tableau	nombre
Vérifier si une personne est majeur grâce à son âge	boolean
Appliquer une réduction de 80% à un prix	nombre
Ajouter un produit a un panier	panier
Supprimer un produit d'un panier	???

XEBIA NATIS

Exemple de besoin	Type de retour
Addition de 2 nombres	nombre
Addition tous les nombres d'un tableau	nombre
Vérifier si une personne est majeur grâce à son âge	boolean
Appliquer une réduction de 80% à un prix	nombre
Ajouter un produit a un panier	panier
Supprimer un produit d'un panier	panier

XWIKI NATIS

Pour construire une fonction, il faut donc se poser 3 questions :

- Type de retour/résultat ?
- Paramètres ou arguments ?
- Algorithme ?

XEBIA NATIS

Exemple de besoin	Paramètres
Addition de 2 nombres	a, b
Addition tous les nombres d'un tableau	tab
Vérifier si une personne est majeur grâce à son âge	???
Appliquer une réduction de 80% à un prix	???
Ajouter un produit a un panier	???
Supprimer un produit d'un panier	???

XEBIA NATIS

Exemple de besoin	Paramètres
Addition de 2 nombres	a, b
Addition tous les nombres d'un tableau	tab
Vérifier si une personne est majeur grâce à son âge	age
Appliquer une réduction de 80% à un prix	???
Ajouter un produit a un panier	???
Supprimer un produit d'un panier	???

XEBIA NATIS

Exemple de besoin	Paramètres
Addition de 2 nombres	a, b
Addition tous les nombres d'un tableau	tab
Vérifier si une personne est majeur grâce à son âge	age
Appliquer une réduction de 80% à un prix	prix
Ajouter un produit a un panier	???
Supprimer un produit d'un panier	???

XDNATIS

Exemple de besoin	Paramètres
Addition de 2 nombres	a, b
Addition tous les nombres d'un tableau	tab
Vérifier si une personne est majeur grâce à son âge	age
Appliquer une réduction de 80% à un prix	prix
Ajouter un produit a un panier	produit, panier
Supprimer un produit d'un panier	???

XEBIA NATIS

Exemple de besoin	Paramètres
Addition de 2 nombres	a, b
Addition tous les nombres d'un tableau	tab
Vérifier si une personne est majeur grâce à son âge	age
Appliquer une réduction de 80% à un prix	prix
Ajouter un produit a un panier	produit, panier
Supprimer un produit d'un panier	produit, panier

XDNATIS

Exemple de besoin	Exemple de déclaration de fonction
Addition de 2 nombres	function additionner(a, b)
Addition tous les nombres d'un tableau	function additionner(tab)
Vérifier si une personne est majeur grâce à son âge	???
Appliquer une réduction de 80% à un prix	???
Ajouter un produit a un panier	???
Supprimer un produit d'un panier	???

XDNATIS

Exemple de besoin	Exemple de déclaration de fonction
Addition de 2 nombres	function additionner(a, b)
Addition tous les nombres d'un tableau	function additionner(tab)
Vérifier si une personne est majeur grâce à son âge	function estMajeur(age)
Appliquer une réduction de 80% à un prix	???
Ajouter un produit a un panier	???
Supprimer un produit d'un panier	???

XDNATIS

Exemple de besoin	Exemple de déclaration de fonction
Addition de 2 nombres	function additionner(a, b)
Addition tous les nombres d'un tableau	function additionner(tab)
Vérifier si une personne est majeur grâce à son âge	function estMajeur(age)
Appliquer une réduction de 80% à un prix	function appliquerReduction(prix)
Ajouter un produit a un panier	???
Supprimer un produit d'un panier	???

XDNATIS

Exemple de besoin	Exemple de déclaration de fonction
Addition de 2 nombres	function additionner(a, b)
Addition tous les nombres d'un tableau	function additionner(tab)
Vérifier si une personne est majeur grâce à son âge	function estMajeur(age)
Appliquer une réduction de 80% à un prix	function appliquerReduction(prix)
Ajouter un produit a un panier	function ajouter(produit, panier)
Supprimer un produit d'un panier	???

XDNATIS

Exemple de besoin	Exemple de déclaration de fonction
Addition de 2 nombres	function additionner(a, b)
Addition tous les nombres d'un tableau	function additionner(tab)
Vérifier si une personne est majeur grâce à son âge	function estMajeur(age)
Appliquer une réduction de 80% à un prix	function appliquerReduction(prix)
Ajouter un produit a un panier	function ajouter(produit, panier)
Supprimer un produit d'un panier	function supprimer(produit, panier)

XWNATIS

Créons une fonction pour trouver le 1er nombre supérieur à 500 dans un tableau :

- Type de retour/résultat : nombre
- Paramètre : un tableau
- Algorithme :

XKNATIS

```
@Fonction chercher @Entrée tab
    resultat <- 0
    @PourChaque element @Dans tab
    @DebutBloc
        @Si element > 500
        @DebutBloc
            resultat <- element
            @Retourner resultat
        @FinBloc
    @FinBloc
```

XWIKI NATIS

TRADUCTION

XKNATIS

```
@Fonction chercher @Entrée tab
    resultat <- 0
    @PourChaque element @Dans tab
    @DebutBloc
        @Si element > 500
        @DebutBloc
            resultat <- element
            @Retourner resultat
        @FinBloc
    @FinBloc
```

XEBIA NATIS

Il y a 3 manières de déclarer une fonction en JS !

```
function chercher(tab)
{
    let resultat = 0;
    for (const a of tab)
    {
        if (a >= 500)
        {
            resultat = a;
            return resultat;
        }
    }
}
```

```
const chercher = function(tab)
{
    let resultat = 0;
    for (const a of tab)
    {
        if (a >= 500)
        {
            resultat = a;
            return resultat;
        }
    }
}
```

```
const chercher = (tab) =>
{
    let resultat = 0;
    for (const a of tab)
    {
        if (a >= 500)
        {
            resultat = a;
            return resultat;
        }
    }
}
```

XEBIA NATIS

Il y a 3 manières de déclarer une fonction en JS !

```
function chercher(tab)
{
    let resultat = 0;
    for (const a of tab)
    {
        if (a >= 500)
        {
            resultat = a;
            return resultat;
        }
    }
}
```

Déclaration conventionnelle d'une fonction avec le mot-clé associé (function)

La syntaxe utilisé est la syntaxe ES5 (avant 2015), bien supporté par l'ensemble des navigateurs

XEBIA NATIS

Il y a 3 manières de déclarer une fonction en JS !

```
const chercher = function(tab)
{
    let resultat = 0;
    for (const a of tab)
    {
        if (a >= 500)
        {
            resultat = a;
            return resultat;
        }
    }
}
```

Déclaration d'une fonction sous forme de constante (variable qui ne change pas de valeur).

Ici, on utilise l'affectation (symbole =) pour mettre une fonction dans cette constante. C'est toujours la syntaxe ES5.

XEBIA NATIS

Il y a 3 manières de déclarer une fonction en JS !

```
const chercher = (tab) =>
{
    let resultat = 0;
    for (const a of tab)
    {
        if (a >= 500)
        {
            resultat = a;
            return resultat;
        }
    }
}
```

Déclaration d'une fonction sous forme de constante (variable qui ne change pas de valeur).

Ici, on utilise l'affectation (symbole =) pour mettre une fonction dans cette constante. C'est toujours la syntaxe ES6 (fonction fléchée).

XEBIA NATIS

Il y a 3 manières de déclarer une fonction en JS !

```
function chercher(tab)
{
    let resultat = 0;
    for (const a of tab)
    {
        if (a >= 500)
        {
            resultat = a;
            return resultat;
        }
    }
}
```

Standard ES5

```
const chercher = function(tab)
{
    let resultat = 0;
    for (const a of tab)
    {
        if (a >= 500)
        {
            resultat = a;
            return resultat;
        }
    }
}
```

Constante ES5

```
const chercher = (tab) =>
{
    let resultat = 0;
    for (const a of tab)
    {
        if (a >= 500)
        {
            resultat = a;
            return resultat;
        }
    }
}
```

Constante ES6

XEBIA NATIS

Exercices de traduction (Fonctions).
Traduire les codes JS en fonctions :
3, 4 & 5

XNATIS

Exercice 3 :

Vérifier si un utilisateur est majeur grâce à son âge. Mettre une variable « resultat » à Vrai si cela est le cas, sinon mettez la à Faux.

Instruction : Age <- 25

Traduction

```
let age = 25;

function estMajeur(age)
{
        if (age >= 18)
        {
           resultat = true;
        }
        else
        {
           resultat = false;
        }
}
```

Traduction

```
let age = 25;

function estMajeur(age)
{
        let resultat = false;
        if (age >= 18)
        {
           resultat =  true;
        }
        return resultat;
}
```

XDNATIS

Exercice 4 :

Vérifier si un utilisateur est mineur grâce à son âge. Mettre une variable « resultat » à Vrai si cela est le cas, sinon mettez la à Faux.

Instruction : Age <- 12

Traduction

```
let age = 12;
let resultat = false;

function estMineur(age)
{
        if (age < 18)       (Ici l'âge est strictement inférieur à 18)
        {
            resultat = true;
        }
        return resultat;
}
```

XDNATIS

Exercice 5 :

Echanger/Inverser les 2 premiers éléments d'un tableau.

Instruction : tab <- [23, 4, 2, 543, 34, 12, 4]

Traduction

```
let tab = [23, 4, 2, 543, 34, 12, 4];

function echanger(tab)
{
        let A = tab[0];
        tab[0] = tab[1];
        tab[1] = A;
        return tab;
}
```

On obtient : tab = [4, 23, 2, 543, 34, 12, 4]

XEBIA NATIS

Exercices de traduction (Fonctions).

Traduire les codes JS en fonctions :

6, 7, 8, 9, 10, 11 & 12

XDNATIS

Exercice 6 :

Compter le nombre d'éléments dans un tableau. Mettre le compte dans une variable "**resultat**".

Instruction : tab <- [23, 4, 2, 543, 34, 12, 4]

Traduction

```
let tab = [23, 4, 2, 543, 34, 12, 4];

function compter(tab)
{
        let resultat = 0;
        for (const element of tab)
        {
                resultat = resultat++ (ou resultat + 1)
        }
        return resultat;
}
```

XDNATIS

Exercice 7.1 :

Faire la somme des éléments d'un tableau.

Instruction : tab <- [23, 4, 2, 543, 34, 12, 4]

Traduction

```
let tab = [23, 4, 2, 543, 34, 12, 4];

function Somme(tab)
{
	let resultat = 0;
	for (const element of tab)
	{
		resultat = resultat + element;
	}
	return resultat;
}
```

XDNATIS

Exercice 7.2 :

Faire la somme des éléments d'un tableau.

Instruction : tab <- [23, 4, 2, 543, 34, 12, 4]

Traduction

```
let tab = [23, 4, 2, 543, 34, 12, 4];

function Somme(tab)
{
        let compte = 0;
        for (const element of tab)
        {
            compte = compte + 1;
        }

        let resultat = 0;
        for (i = 0; i < compte; i++)
        {
           resultat = resultat + tab[i] ;
        }
        return resultat;
}
```

XNATIS

Exercice 8 :

Trouver l'élément maximum d'un tableau. Mettre le nombre maximum dans une variable "**resultat**".

Instruction : tab <- [23, 4, 2, 543, 34, 12, 4]

Traduction

```
let tab = [23, 4, 2, 543, 34, 12, 4];

function Maximum(tab)
{
        let resultat = 0;
        for (const element of tab)
            {
                if (element > resultat)
                    {
                        resultat = element;
                    }
            }
            return resultat;
}
```

XDNATIS

Exercice 9 :

Trouver l'élément minimum d'un tableau. Mettre le nombre maximum dans une variable "**resultat**".

Instruction : tab <- [23, 4, 2, 543, 34, 12, 4]

Traduction

```
let tab = [23, 4, 2, 543, 34, 12, 4];

function Minimum(tab)
{
        let resultat = 0;
        for (const element of tab)
           {
               if (element <resultat)
                  {
                      resultat = element;
                  }
           }
           return resultat;
}
```

XDNATIS

Exercice 10.1 :

Trouver le premier élément supérieur à 500. S'il n'y en a pas,
le résultat doit être 0.

Instruction : tab <- [23, 4, 2, 543, 34, 12, 4]

Traduction (Méthode 1 : for… of)

```
let tab = [23, 4, 2, 543, 34, 12, 4];

function Sup500(tab)
{
        let resultat = 0;

        for (const element of tab)                    (Pour chaque élément du tableau parcouru)
          {
              if (element > 500)
                {
                  resultat = element;
                  break;                              (On termine la boucle avec l'instruction « Break »)
                }
          }
}
```

XDNATIS

Exercice 10.2 :

Trouver le premier élément supérieur à 500. S'il n'y en a pas, le resultat doit être 0.

Instruction : tab <- [23, 4, 2, 543, 34, 12, 4]

Traduction (Méthode 2 : while)

```
function Sup500(tab)
{
        let i = 0;
        let indiceMax = tab.length -1;

        while (tab[i] < 500 && i <= indiceMax) (Tant que l'élément à la position actuelle est strictement inférieure à 500 et
        que i ne dépasse pas la position maximale autorisée c'est que l'on a pas encore trouvé ce que l'on cherche, on peut
        avancer)
        {
            i = i + 1;                      (On continue d'avancer)
            if (i > indiceMax)              (Si la position actuelle est strictement supérieure à la position maximale autorisée)
              {
                 break;                     (On termine la boucle avec l'instruction « Break »)
              }
        }
}
```

XDNATIS

Exercice 10.3 :

Trouver le premier élément supérieur à 500. S'il n'y en a pas, le résultat doit être 0.

Instruction : tab <- [23, 4, 2, 543, 34, 12, 4]

Traduction (Méthode 3 : For)

```
function Sup500(tab)
{
        let resultat = 0;
        If (tab[i] >= 500)                    (Si l'on trouve ce que l'on cherche)
                {
                    resultat = tab[i];        (On affecte la valeur trouvée à resultat)
                }
        return resultat;
}
```

XNATIS

Exercice 11.1 :

Copier le tableau "tab" dans un autre tableau intitulé "autre".

Instruction : tab <- [23, 4, 2, 543, 34, 12, 4]

Traduction (Méthode 1)

```
let tab = [23, 4, 2, 543, 34, 12, 4];

function copierTableau(tab)
{
        let autre = [];
        for (const element of tab)
                {
                    autre.push(element);
                }
        return autre;
}
```

(On ajoute « push » chaque élément un par un à ce nouveau tableau « autre »)

XNATIS

Exercice 11.2 :

Copier le tableau "tab" dans un autre tableau intitulé "autre".

Instruction : tab <- [23, 4, 2, 543, 34, 12, 4]

Traduction (Méthode 2)

```
let tab = [23, 4, 2, 543, 34, 12, 4];
function copierTableau(tab)
{
        let autre = [];                          (On créé un nouveau tableau « autre »)

        for (const element of tab)
                {
                    autre.push(element);         (On ajoute chaque élément un par un à ce nouveau tableau « autre » en utilisant « push »)
                }
}
```

XNATIS

Exercice 12 :

Copier les premiers éléments d'un tableau dont la somme fait au moins 500 (dans un autre tableau intitulé « autre »).

Instruction : tab <- [23, 4, 2, 543, 34, 12, 4]

Traduction

```
function copierElementSomme500(tab)
{
    let autre = [];
    let somme = 0;
    for (const element of tab)
        {
            somme = somme + element;
            autre.push(element);
            if (somme >= 500)
                {
                break;
                }
        }
        return autre;
}
```