



Les fondamentaux de la programmation Algorithmie

cci
campus
ALSACE

Yoann VINCENT

Présentation

Qui êtes-vous ?

- Yoann VINCENT – 28 ans
- Développeur – Formateur en informatique



Où travaillez-vous ?

- Auto-entrepreneur : VYDev
- Intervenant dans diverses entreprises

Et vous, qui êtes-vous ?

- Prénom, nom
- Entreprise, mission / poste occupez
- Objectif de carrière



...

Objectifs du cours



Traduire en pseudo-code un traitement, une suite d'actions



Maîtriser les structures algorithmiques



Produire et comprendre des algorithmes



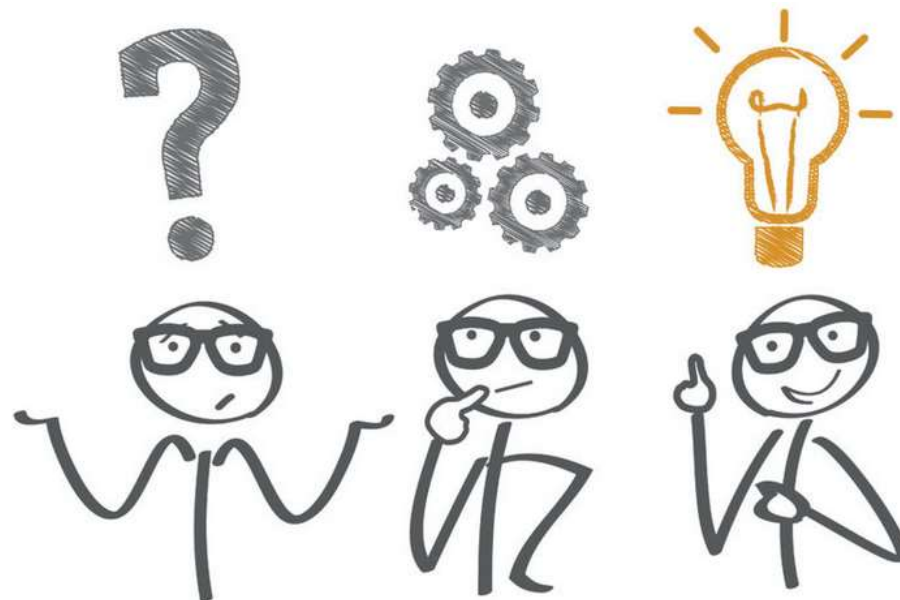
1. Introduction

1. Introduction

Rappel :

Qu'est-ce qu'un algorithme ?

Réponse : Un algorithme est une suite logique d'instruction permettant de donner la réponse à un problème



1. Introduction

Le HTML c'est quoi ?

Hyper Text Markup Language est le code utilisé pour structurer une page web.



Le CSS c'est quoi ?

Cascading Style Sheets est le code utilisé pour la mise en forme du contenu.




```

10
11
12
13
14
15
16
17
18
19
20
21
<link rel="stylesheet" href="http://localhost/css.css" type="text/css">
<script type="text/javascript" src="http://localhost/javascript.js"></script>
<script type="text/javascript">
(function(){
  onLoaded: function(request) {
    if (request.name == 'log_error') return;
    log.trace("Ajax.Request: " + (request.name || request.url.substr(0, 30)) + "...");
  },
  onComplete: function(request) {
    if (request.name == 'log_error') return;
  },
  onException: function(request, e) {
    if (request.name == 'log_error') return;
    log.fatal(request.url + ': ' + e.name + ' | ' + e.message + ' | ' +
      e.stack);
  }
});

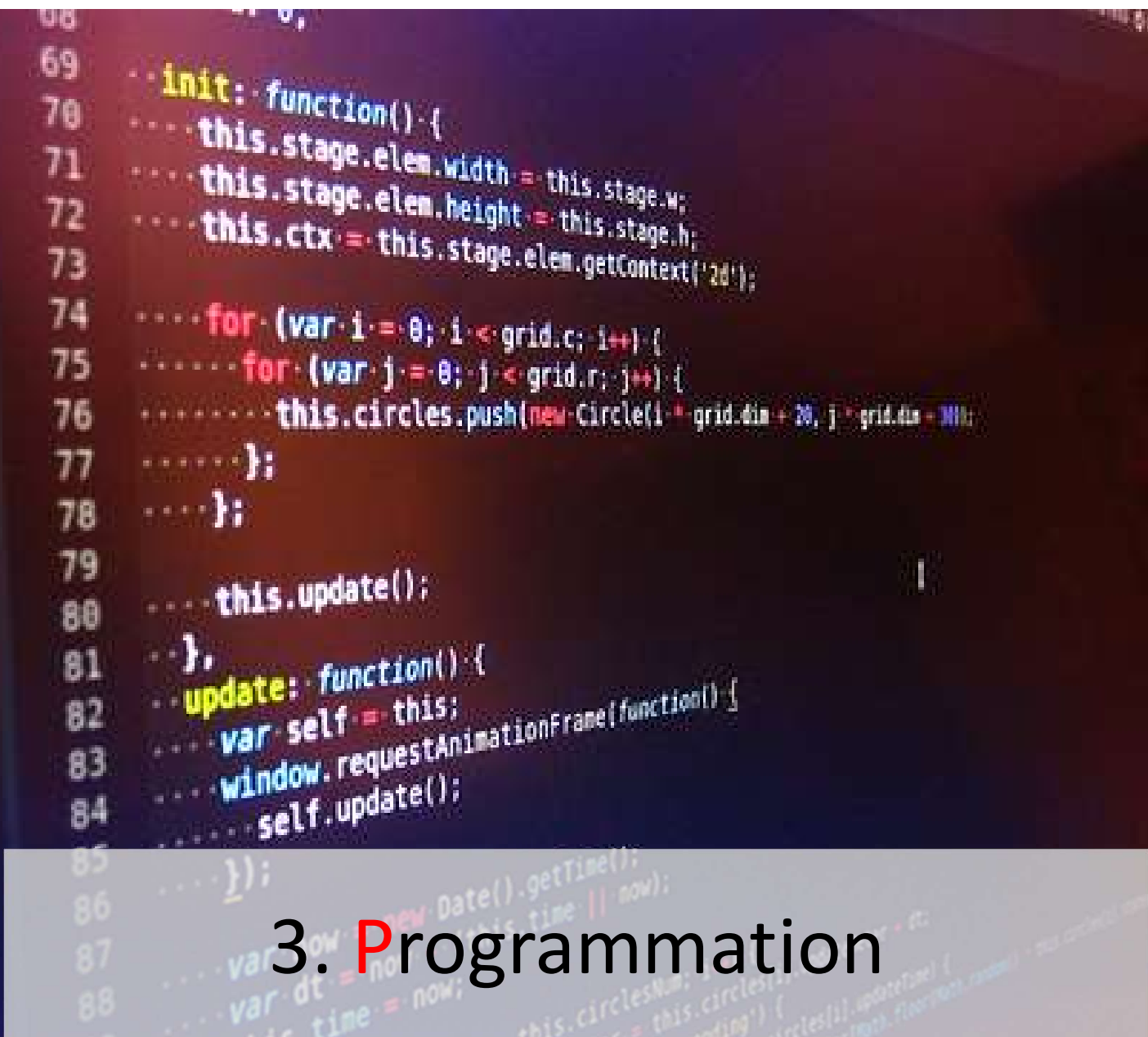
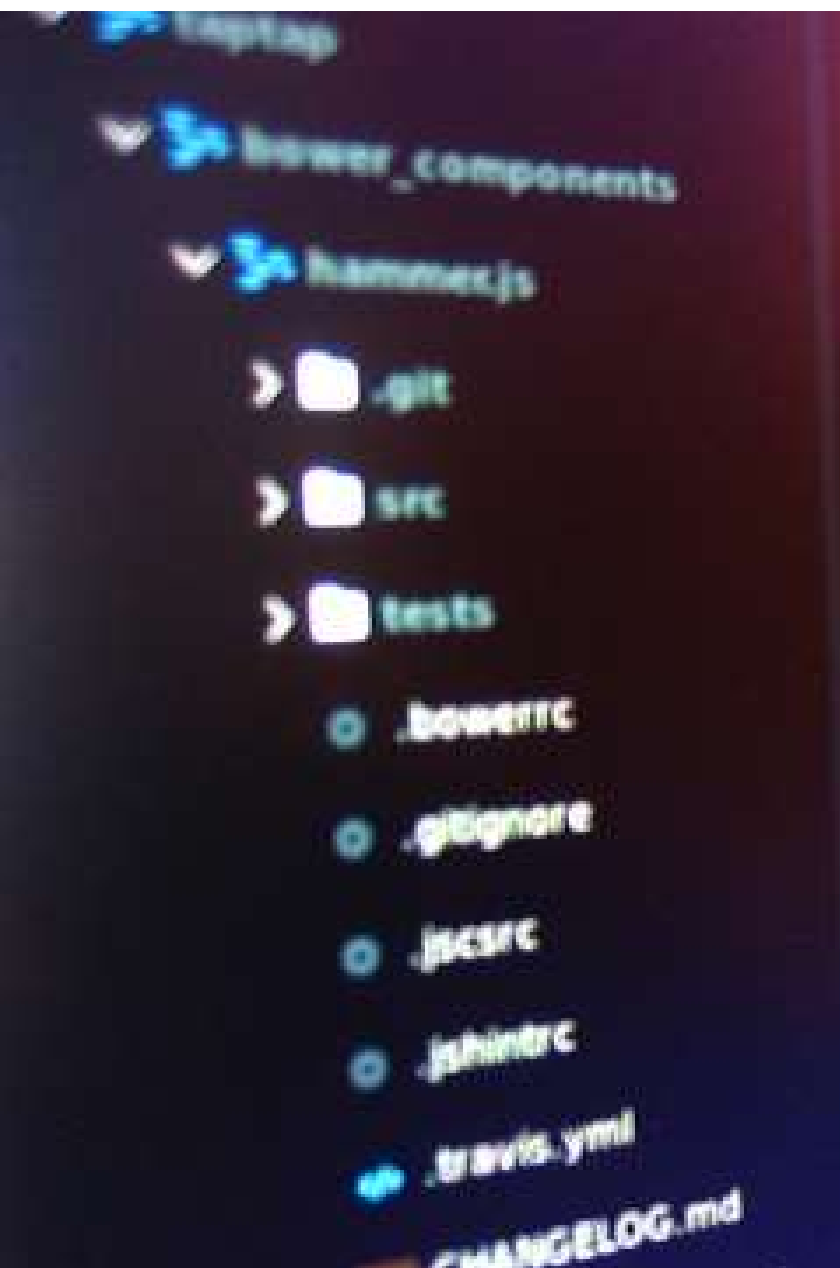
```

2. JavaScript

2. JavaScript

Le JavaScript c'est quoi ?

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web avec les langages HTML et CSS.



3. Programmation

3.1 **P**our commencer

Intégration du JS dans une page HTML / CSS :

1^{er} méthode :

```
<script>  
...  
</script>
```

2^{ème} méthode :

```
<script src="..."></script>
```

Méthode recommandée

3.2 **C**ommentaire et terminaison

Écrire un commentaire :

Sur une ligne : //

Sur plusieurs lignes : /* */

Exemple :

// Je suis un commentaire

/* Bonjour

* Nous sommes des commentaires */

Terminaison de ligne :

Les lignes de code termine par un **point virgule**

Exemple :

alert("Bonjour"); 

3.3 Les variables

Convention de nommage :

Le plus explicite possible

Commence par une minuscule et jamais par un chiffre

Ne comporte pas de caractère spéciaux ni d'espace

Si le nom comporte plusieurs mots, la première lettre sera en majuscule (camel case)

Eviter d'utiliser les mots réservés : var, function, for, ...

Exemple :

ageUtilisateur

nomUtilisateur

date

moyenne

3.3 Les variables

Les variables JavaScript ne sont pas typés, il est conseillé d'indiquer le type dans le nom de la variable.

Les variables ne doivent pas être des mots réservés : var, form, document, ...

Déclaration de variable :

```
var stringName;
```

```
var bool;
```

```
var intAge;
```

Initialisation:

```
var stringName;
```

```
stringName = "Yoann";
```

```
var stringName = "Yoann";
```

```
var bool;
```

```
bool = true;
```

```
var bool = true;
```

```
var intAge;
```

```
intAge = 30;
```

```
var intAge = 30;
```

3.4 Affectations

Affectation simple : **=**

Permet de définir la valeur d'une variable

Exemple : `var number = 2;`

Affectation / addition : **+=**

Permet de calculer la somme ou la concaténation de ses deux opérandes

Exemple :

```
var number = 2;  
number += 1;
```

Exemple :

```
var name = "Jean";  
name += " dupont";
```


3.4 Affectations

Affectation / soustraction : **`-=`**

Permet de calculer la soustraction de ses deux opérandes

Exemple :

```
var number = 2;  
number -= 1;
```

Affectation / multiplication : **`*=`**

Permet de calculer la multiplication de ses deux opérandes

Exemple :

```
var number = 2;  
number *= 1;
```

3.4 Affectations

Affectation / division : **/=**

Permet de calculer la division de ses deux opérandes

Exemple :

```
var number = 2;  
number /= 2;
```

3.5 Entrées / sorties

Entrées (lire) :

```
prompt("Quel est votre nom ?");
```

```
var name = prompt("Quel est votre nom ?");
```

Sorties (écrire) :

```
alert("Bonjour");
```

```
alert("Bonjour " + name);
```



Visual Studio Code

3.6 | Introduction Visual Studio code + Exercices

3.6 Exercices

Exercice 1 :

Créer et initialiser 3 variables puis afficher leurs contenu sur 3 lignes différentes.
Commenter votre code. Respecter le camel case

Résultat attendu :



3.6 Exercices

Exercice 1 solution :

```
var stringName = "Yoann";  
var intNumber = 30;  
var bool = true;  
// Affichage des variables  
alert(stringName + "\n" + intNumber + "\n" + bool);
```

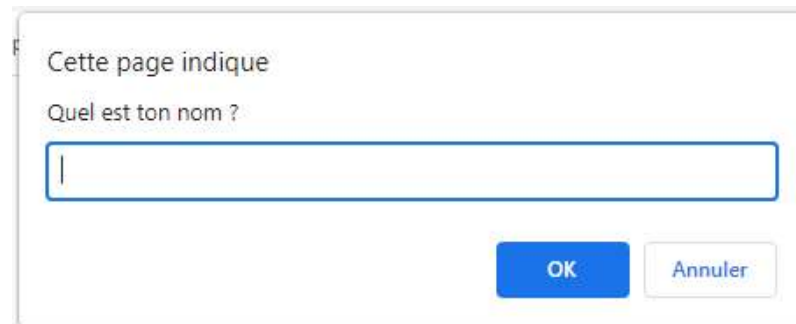

3.6 Exercices

Exercice 2 :

Demander le nom, puis écrire le résultat.

Commenter votre code. Respecter le camel case

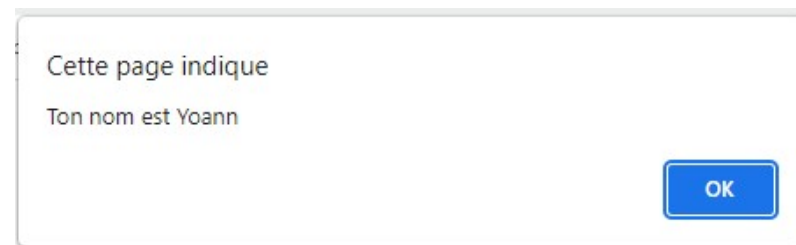
Résultat attendu :



Cette page indique

Quel est ton nom ?

OK Annuler



Cette page indique

Ton nom est Yoann

OK

3.6 Exercices

Exercice 2 solution :

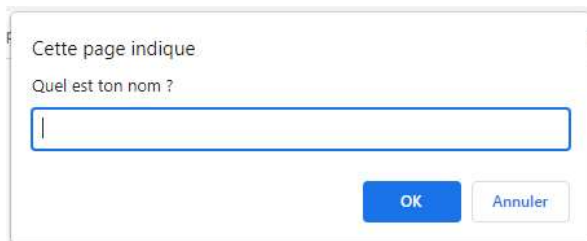
```
var stringName = prompt("Quel est ton nom ?");  
alert("Ton nom est " + stringName);
```

3.6 Exercices

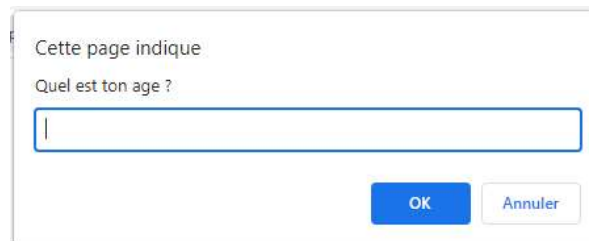
Exercice 3 :

Demander le nom ainsi que l'âge, puis écrire le résultat.
Commenter votre code. Respecter le camel case

Résultat attendu :



Cette page indique
Quel est ton nom ?



Cette page indique
Quel est ton âge ?



Cette page indique
Ton nom est Yoann et tu as 28 ans

3.6 Exercices

Exercice 3 solution :

```
var stringName = prompt("Quel est ton nom ?");  
var intAge = prompt("Quel est ton age ?");  
alert("Ton nom est " + stringName + " et tu as " + intAge + " ans");
```

```
var stringName = prompt("Quel est ton nom ?");  
var intAge = prompt("Quel est ton age ?");  
alert(`Ton nom est ${stringName} et tu as ${intAge} ans`);
```

```
alert(`Ton nom est ${prompt("Quel est ton nom ?")} et tu as ${prompt("Quel est ton age ?")} ans`);
```

3.7 **C**alculs

Opérateur :

+ - / * %

	var multiplication = a * b;	var multiplication = 3 * 5;
var a = 3;	var addition = a + b;	var addition = 3 + 5;
	var soustraction = a - b;	var soustraction = 3 - 5;
var b = 5;	var division = a / b;	var division = 3 / 5;
	var modulo = a % b;	var modulo = 3 % 5;

Fonction : Number(a)

3.7 C Calculs

Exercice :

Quelle est la valeur de la variable C à la fin du programme ?

```
var a = 3 ;  
var b = 5;  
var c = 7;
```

```
c = a + b;
```

Réponse : c = 35

3.8 Condition

Syntaxe :

```
if(...)
{
    ...
}
else
{
    ...
}
```

Opérateur de comparaison :

<	Inférieur à	&&	Et logique
>	Supérieur à		Ou logique
<=	Inférieur ou égal à	!	Négation
>=	Supérieur ou égal à		
==	Égal		
!=	Différent de		

3.8 Condition

Exemple :

```
var a = 3;  
var b = 5;  
var c = 7;
```

```
if( a > b)  
{  
    ...  
}  
else  
{  
    ...  
}
```

```
var a = 3;  
var b = 5;  
var c = 7;
```

```
if( a == b)  
{  
    ...  
}  
else  
{  
    ...  
}
```

```
var a = 3;  
var b = 5;  
var c = 7;
```

```
if( a <= b)  
{  
    ...  
}  
else  
{  
    ...  
}
```

3.8 Condition

Multiple condition :

Possibilité de faire de multiple condition en une fois

```
if (( ... ) && ( ... ))  
{  
    ...  
}  
else  
{  
    ...  
}
```

```
if (( ... ) || ( ... ))  
{  
    ...  
}  
else  
{  
    ...  
}
```

3.8 Condition

Exemple :

```
var a = 3;  
var b = 5;  
var c = 7;
```

```
if (( a > b ) && ( a > c ))  
{  
    ...  
}  
else  
{  
    ...  
}
```

```
var a = 3;  
var b = 5;  
var c = 7;
```

```
if (( a != b ) || ( c == b ))  
{  
    ...  
}  
else  
{  
    ...  
}
```

3.8 Condition

Switch / case :

Le switch case est une condition multiple qui permet de remplacer des if a la suite

Le switch case **regarde si la condition est vrai**

Syntaxe :

```
switch (VARIABLE)
{
    case CONDITION:
        ...
        break;
    case CONDITION:
        ...
        break;
    ...
}
```

3.8 Condition

Exemple avec condition if :

```
var jourSemaine = "Lundi";
if (jourSemaine == "Lundi")
{
    alert("Nous sommes lundi");
}
else if (jourSemaine == "Mardi")
{
    alert("Nous sommes mardi");
}
else if (jourSemaine == "Mercredi")
{
    alert("Nous sommes mercredi");
}
else
{
    alert("Un jour de la semaine");
}
```

Exemple avec condition switch / case :

```
var jourSemaine = "Lundi";
switch (jourSemaine)
{
    case "Lundi":
        alert("Nous sommes lundi");
        break;
    case "Mardi":
        alert("Nous sommes mardi");
        break;
    case "Mercredi":
        alert("Nous sommes mercredi");
        break;
    default:
        alert("Un jour de la semaine");
        break;
}
```


3.9 Exercices

Exercice 4 :

Demander l'âge et indiquer si la personne est âgée (ou pas)(limite 50 non inclus), puis écrire le résultat.

Commenter votre code. Respecter le camel case

Résultat attendu :

The image displays three dialog boxes illustrating the expected result of the exercise:

- Top Dialog:** Titled "Cette page indique", it contains the question "Quel age as-tu ?" followed by a text input field. At the bottom, there are two buttons: "OK" and "Annuler".
- Bottom Left Dialog:** Titled "Cette page indique", it displays the message "Tu es jeune !" and has an "OK" button.
- Bottom Right Dialog:** Titled "Cette page indique", it displays the message "Tu es vieux !" and has an "OK" button.

3.9 Exercices

Exercice 4 solution :

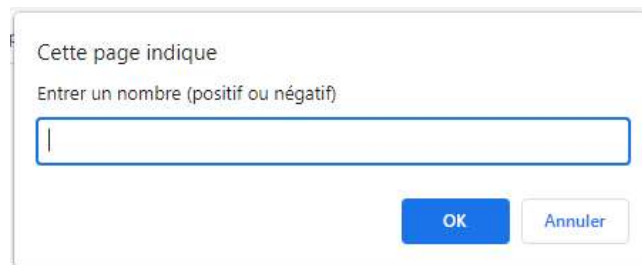
```
var intAge = prompt("Quel age as-tu ?");  
// Condition  
if (intAge > 50) {  
    alert("Tu es vieux !");  
} else {  
    alert("Tu es jeune !");  
}
```

3.9 Exercices

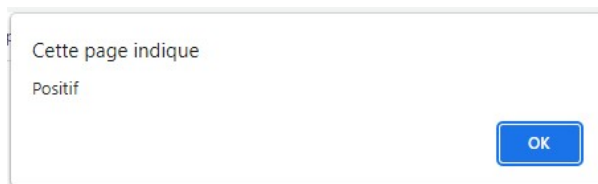
Exercice 5 :

Demander un nombre et indiquer si le nombre est positif ou négatif
Commenter votre code. Respecter le camel case

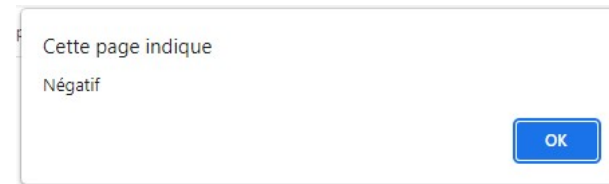
Résultat attendu :



A dialog box titled "Cette page indique" with the prompt "Entrer un nombre (positif ou négatif)". It features a text input field and two buttons at the bottom: "OK" (blue) and "Annuler" (light gray).



A dialog box titled "Cette page indique" showing the result "Positif". It has a single "OK" button (blue) at the bottom right.



A dialog box titled "Cette page indique" showing the result "Négatif". It has a single "OK" button (blue) at the bottom right.

3.9 Exercices

Exercice 5 solution :

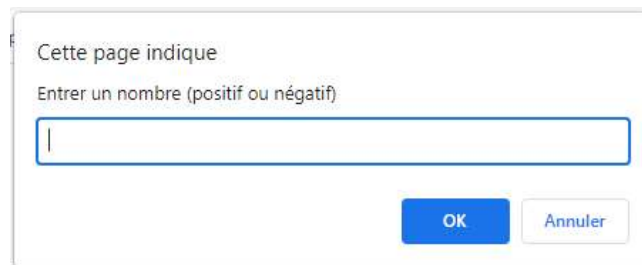
```
var number = prompt("Entrer un nombre (positif ou négatif)");  
// Condition  
if (number > 0) {  
    alert("Positif");  
} else {  
    alert("Négatif");  
}
```

3.9 Exercices

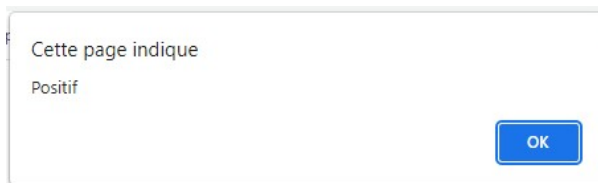
Exercice 6 :

Demander un nombre et indiquer si le nombre est positif, négatif ou égal à 0
Commenter votre code. Respecter le camel case

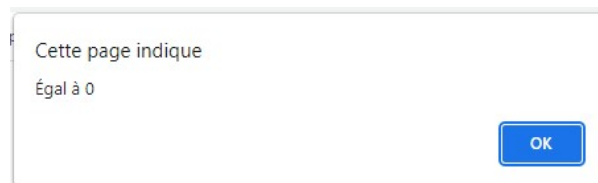
Résultat attendu :



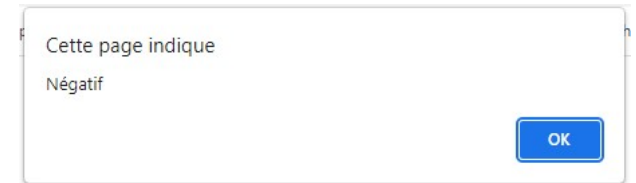
A dialog box titled "Cette page indique" with the prompt "Entrer un nombre (positif ou négatif)". It features a text input field and two buttons at the bottom: "OK" (blue) and "Annuler" (light gray).



A dialog box titled "Cette page indique" with the message "Positif". It has a single "OK" button (blue) at the bottom right.



A dialog box titled "Cette page indique" with the message "Égal à 0". It has a single "OK" button (blue) at the bottom right.



A dialog box titled "Cette page indique" with the message "Négatif". It has a single "OK" button (blue) at the bottom right.

3.9 Exercices

Exercice 6 solution :

```
var number = prompt("Entrer un nombre (positif, négatif ou égal à 0)");  
// Condition  
if (number > 0) {  
    alert("Positif");  
} else if (number == 0){  
    alert("Égal à 0");  
} else {  
    alert("Négatif");  
}
```

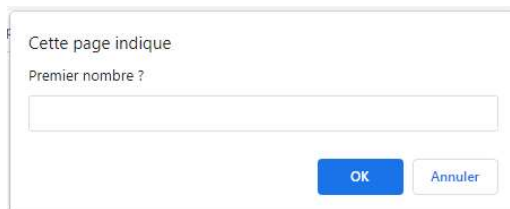
3.9 Exercices

Exercice 7 :

Demander deux nombre et un opérateur (+ ou – ou * ou /) puis effectuer le calcul et afficher le résultat. Structure if imposé.

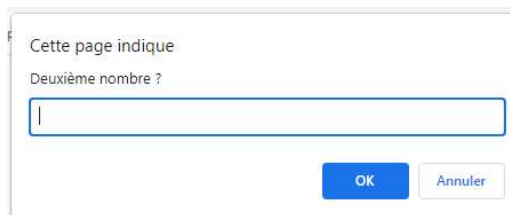
Commenter votre code. Respecter le camel case

Résultat attendu :




Cette page indique
Premier nombre ?

OK Annuler



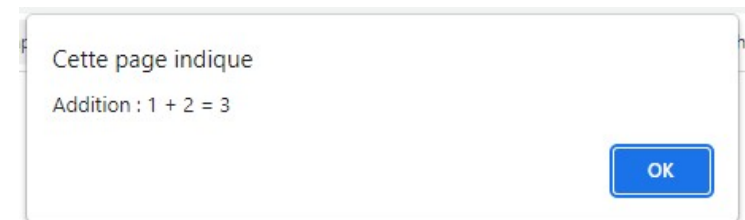
Cette page indique
Deuxième nombre ?

OK Annuler



Cette page indique
Opérateur ? (+ OU - OU * OU /)

OK Annuler



Cette page indique
Addition : 1 + 2 = 3

OK

3.9 Exercices

Exercice 7 solution :

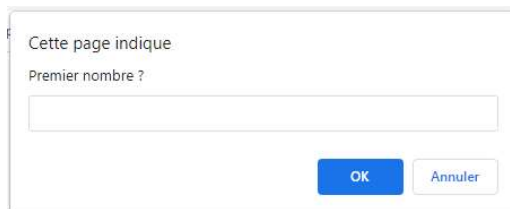
```
var firstNumber = prompt("Premier nombre ?");
var secondNumber = prompt("Deuxième nombre ?");
var operator = prompt("Opérateur ? ( + OU - OU * OU /)");
// Condition
if (operator == '+') {
    alert(`Addition : ${firstNumber} + ${secondNumber} = ${Number(firstNumber) + Number(secondNumber)}`);
} else if (operator == '-') {
    alert(`Soustraction : ${firstNumber} - ${secondNumber} = ${Number(firstNumber) - Number(secondNumber)}`);
} else if (operator == '*') {
    alert(`Multiplication : ${firstNumber} * ${secondNumber} = ${Number(firstNumber) * Number(secondNumber)}`);
} else if (operator == '/') {
    alert(`Division : ${firstNumber} / ${secondNumber} = ${Number(firstNumber) / Number(secondNumber)}`);
} else {
    alert('Opérateur non reconnu');
}
```


3.9 Exercices

Exercice 8 :

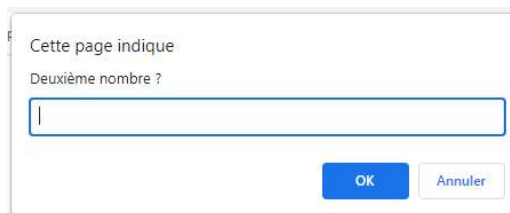
Demander deux nombre et un opérateur (+ ou – ou * ou /) puis effectuer le calcul et afficher le résultat. Structure **switch/case** imposé.
Commenter votre code. Respecter le camel case

Résultat attendu :



Cette page indique
Premier nombre ?

OK Annuler



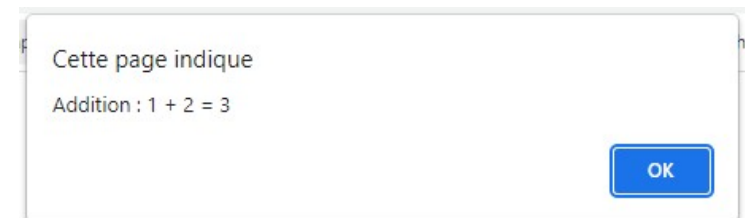
Cette page indique
Deuxième nombre ?

OK Annuler



Cette page indique
Opérateur ? (+ OU - OU * OU /)

OK Annuler



Cette page indique
Addition : 1 + 2 = 3

OK

3.9 Exercices

Exercice 8 solution :

```
var firstNumber = prompt("Premier nombre ?");
var secondNumber = prompt("Deuxième nombre ?");
var operator = prompt("Opérateur ? ( + OU - OU * OU / )");
switch (operator){
    case '+' :
        alert(`Addition : ${firstNumber} + ${secondNumber} = ${Number(firstNumber) + Number(secondNumber)}`);
        break;
    case '-' :
        alert(`Soustraction : ${firstNumber} - ${secondNumber} = ${Number(firstNumber) - Number(secondNumber)}`);
        break;
    case '*' :
        alert(`Multiplication : ${firstNumber} * ${secondNumber} = ${Number(firstNumber) * Number(secondNumber)}`);
        break;
    case '/' :
        alert(`Division : ${firstNumber} / ${secondNumber} = ${Number(firstNumber) / Number(secondNumber)}`);
        break;
    default:
        alert('Opérateur non reconnu');
}
```

3.10 **B**oucles

Boucle while :

La boucle while permet d'effectuer plusieurs fois le même code. Tant que la condition **d'entrée** est vraie, on boucle.

Syntaxe :

```
while(CONDITION)
{
    ...
}
```

Exemple :

```
var compte = 0;
while (compte < 5)
{
    alert("Je suis passé par la ! ");
    compte++;
}
```

3.10 **B**oucles

Boucle do/while :

La boucle do/while permet d'effectuer plusieurs fois le même code. Tant que la condition de **sortie** est vraie, on boucle.

Syntaxe :

```
do
{
    ...
}
while(CONDITION)
```

Exemple :

```
var chiffre = 0;
do
{
    alert("Et par la aussi ! ");
    chiffre++;
} while (chiffre < 5);
```

3.10 **B**oucles

Boucle for :

La boucle for permet d'effectuer plusieurs fois le même code tant que la condition est vrai

Syntaxe : for (INITIALISATION;CONDITION;INCREMENTEUR)
 {
 ...
 }

Exemple : for (var i = 0; i < 5; i++)
 {
 alert("C'est encore moi");
 }

3.11 Exercices

Exercice 9 :

Écrire le décompte de 0 à 3. Structure **boucle do/while** imposé.
Commenter votre code. Respecter le camel case

Résultat attendu :

Cette page indique
0

OK

Cette page indique
1

OK

Cette page indique
2

OK

Cette page indique
3

OK

3.11 Exercices

Exercice 9 solution :

```
var number = 0;
do
{
    alert(number);
    number++;
} while (number < 4);
```

3.11 Exercices

Exercice 10 :

Écrire le décompte de 0 à 3. Structure boucle while imposé.
Commenter votre code. Respecter le camel case

Résultat attendu :

Cette page indique
0

OK

Cette page indique
1

OK

Cette page indique
2

OK

Cette page indique
3

OK

3.11 Exercices

Exercice 10 solution :

```
var number = 0;
while (number < 4)
{
    alert(number);
    number++;
}
```

3.11 Exercices

Exercice 11 :

Écrire le décompte de 0 à 3. Structure boucle for imposé.
Commenter votre code. Respecter le camel case

Résultat attendu :

Cette page indique
0

OK

Cette page indique
1

OK

Cette page indique
2

OK

Cette page indique
3

OK

3.11 Exercices

Exercice 11 solution :

```
for (var i = 0; i < 4; i++)  
{  
    alert(i);  
}
```

3.11 Exercices

Exercice 12 :

Demander un chiffre puis écrire un message le nombre de fois demander.
Structure **boucle do/while, boucle while et boucle for**, imposé.
Commenter votre code. Respecter le camel case

Résultat attendu :

The diagram illustrates the expected output for Exercise 12, showing three different loop structures used to prompt for a number and display a message a specific number of times. Each dialog box has a title 'Cette page indique' and an 'OK' button.

- do/while loop:** Two dialog boxes. The first says 'Boucle do/while : 1' and the second says 'Boucle do/while : 2'.
- while loop:** Two dialog boxes. The first says 'Boucle while : 1' and the second says 'Boucle while : 2'.
- for loop:** Two dialog boxes. The first says 'Boucle for : 1' and the second says 'Boucle for : 2'.

Additionally, a larger dialog box at the top center shows the input prompt: 'Donne moi un chiffre ?' with a text field containing the number '2' and buttons for 'OK' and 'Annuler'.

3.11 Exercices

Exercice 12 solution :

```
var number = prompt("Donne moi un chiffre ?");

var account = 0;

// Boucle do/while
do
{
    alert(`Boucle do/while : ${Number(account) + 1}`);
    account++;
} while (account < number);

account = 0;

// Boucle while
while (account < number)
{
    alert(`Boucle while : ${Number(account) + 1}`);
    account++;
}

// Boucle for
for (var i = 0; i < number; i++)
{
    alert(`Boucle for : ${Number(i) + 1}`);
}
```

3.12 Les fonctions

Définition :

Une fonction est un ensemble de lignes de code réalisant des actions.

Chaque fonction est réutilisable à l'infini et peut être paramétrée avec des arguments.

Une fonction peut :

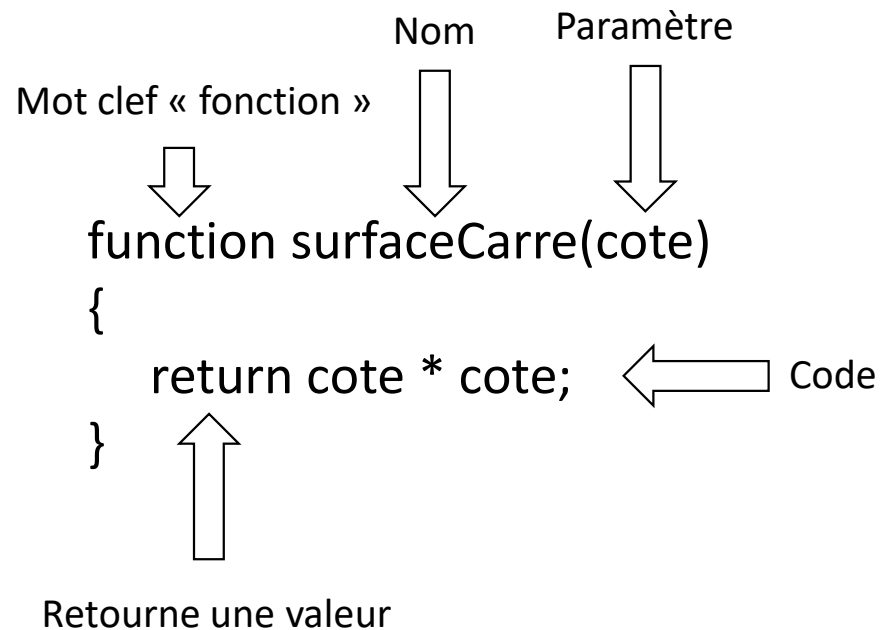
- Retourner ou non une valeur
- Admettre ou non un argument

Un argument c'est quoi ?

Argument aussi appelé paramètre va permettre de passer des informations à la fonction sous forme de variable ou d'objet.

3.12 Les fonctions

Exemple :



3.14 Sélécteur

Définition :

Un sélecteur permet de cibler un élément afin de pouvoir interagir avec.

Plusieurs sélecteurs existent :

- `getElementById('...')` : Permet de sélectionner par l'**attribut id** de la balise HTML
- `getElementsByName('...')` : Permet de sélectionner par l'**attribut name** de la balise HTML
- `getElementsByTagName('...')` : Permet de sélectionner par le **nom** de la balise HTML
- `getElementsByClassName('...')` : Permet de sélectionner par l'attribut class de la balise HTML

Chaque sélecteur est précédé du mot « document »

Exemple : `document.getElementById('...');`

3.14 Sélécteur

Exemple :

HTML :

```
...  
<input type="text" name="name" id="name" />  
...
```

JS :

```
...  
var stringName = document.getElementById('name').value;  
...
```

3.15 Événements

Définition :

Un événement est une réaction suite à une action émise par l'utilisateur, comme un clique, survol de la souris, ...

Plusieurs événements existent :

- Onclick : Lors du clique de l'utilisateur
- Onload : Lorsque la page est chargée
- Onmouseout : Lorsque la souris sors de l'élément
- Onmouseover : Lorsque la souris est sur l'élément
- ...

Ils peuvent être insérés dans les balises HTML.

3.15 Événements

Exemple :

HTML :

```
<input type="text" name="name" id="name" />  
<input type="button" onclick="send()" />
```

JS :

```
function send()  
{  
    var stringName = document.getElementById('name').value;  
    alert(stringName);  
}
```

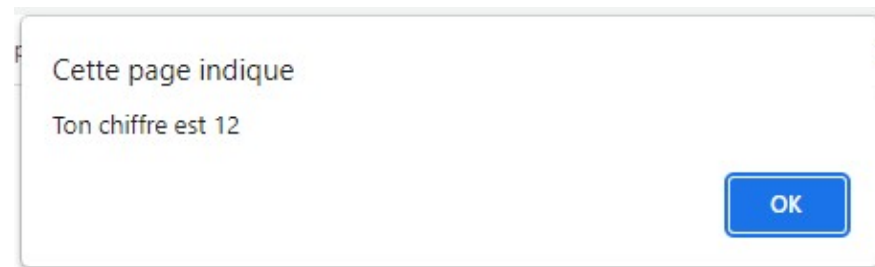
3.16 Exercices

Exercice 13 :

Demander un chiffre dans un formulaire HTML puis écrire le chiffre dans une alerte JS
Commenter votre code. Respecter le camel case

Résultat attendu :

Chiffre



3.16 Exercices

Exercice 13 solution :

```
<form action="" method="get">
  <label for="number">Chiffre</label>
  <input type="text" name="number" id="number"/>
  <input type="button" value="Envoyer" onclick="send()">
</form>
```

```
function send()
{
  var number = document.getElementById('number').value;
  alert(`Ton chiffre est ${number}`);
}
```

3.16 Exercices

Exercice 14 :

Demander un chiffre dans un formulaire HTML puis afficher le carré (2) du chiffre dans une alerte JS

Commenter votre code. Respecter le camel case

Résultat attendu :

Chiffre au carré



3.16 Exercices

Exercice 14 solution :

```
<form action="" method="get">
  <label for="number">Chiffre au carré</label>
  <input type="text" name="number" id="number"/>
  <input type="button" value="Envoyer" onclick="send()">
</form>
```

```
function send()
{
  var number = document.getElementById('number').value;
  alert(`Ton chiffre au carrée est ${Number(number*number)}`);
}
```

3.16 Exercices

Exercice 15 :

Réaliser le formulaire suivant. Afficher le nom prénom et si l'utilisateur est âgée (ou pas) (limite 50 non inclus).

Commenter votre code. Respecter le camel case

Résultat attendu :

Nom

Prénom

Age

Cette page indique

Ton nom est VINCENT | Ton prénom est Yoann | Tu es vieux !

3.16 Exercices

Exercice 15 solution :

```
<form action="" method="get">
  <div>
    <label for="name">Nom</label>
    <input type="text" name="name" id="name">
  </div>
  <div>
    <label for="firstName">Prénom</label>
    <input type="text" name="firstName" id="firstName">
  </div>
  <div>
    <label for="age">Age</label>
    <input type="number" name="age" id="age">
  </div>
  <div>
    <input type="button" value="Envoyer" onclick="send()">
  </div>
</form>
```

```
function send()
{
  var stringName = document.getElementById('name').value;
  var stringFirstName = document.getElementById('firstName').value;
  var age = document.getElementById('age').value;

  var response = `Ton nom est ${stringName} | Ton prénom est ${stringFirstName} | `;

  if(age > 50)
  {
    response += "Tu es vieux !";
  }
  else
  {
    response += "Tu es jeune !";
  }
  alert(response);
}
```

3.17 Tableaux

Définition :

Un tableau est une manière de stocker des données ressemblant a une liste.

Le type des éléments sont pas fixés

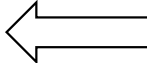
Syntaxe :

Nom du tableau
↓
Création du tableau
↓
`var myTable = new Array(...);`
↑
Permet de définir le nombre d'élément du tableau.
Paramètre non obligatoire

3.17 Tableaux

Alimenter un tableau :

```
var myTable = new Array("Bonjour" , "Salut" , "Hello");
```

```
var myTable = new Array(3);  3 éléments dans le tableau : 0, 1, 2  
myTable = ["Bonjour" , "Salut" , "Hello"];
```

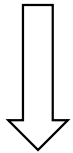
```
var myTable = new Array(3);  
myTable[0] = "Bonjour";  
myTable[1] = "Salut";  
myTable[2] = "Hello";
```

3.17 Tableaux

Exemple :

```
var myTable = new Array("Bonjour" , "Salut" , "Hello");
```

```
alert(myTable[1]);
```



Salut

myTable →

Bonjour →

Salut →

Hello →



3.17 Tableaux

Parcourir un tableau :

Boucle FOR

```
for (INITIALISATION;CONDITION;INCREMENTEUR)
{
    ...
}
```

Propriété length :

Propriété permettant de renvoyer le nombre d'éléments d'un tableau

3.17 Tableaux

Exemple :

```
var myTable = new Array("Bonjour" , "Salut" , "Hello");  
  
for (var index = 0; index < myTable.length; index++)  
{  
    alert(myTable[index]);  
}
```

3.17 Tableaux

Parcourir un tableau :

Boucle FOREACH

Définition :

Permet de parcourir le tableau élément par élément

Syntaxe :

Nom du tableau Désigne un élément

 Boucle forEach

```
myTable.forEach(element =>
{
    ...
});
```

← Code

The diagram illustrates the syntax of the `foreach` loop. It shows the code `myTable.forEach(element => { ... });`. Arrows point from the labels 'Nom du tableau' to `myTable`, 'Boucle forEach' to `forEach`, and 'Désigne un élément' to `element`. A horizontal arrow points from the word 'Code' to the opening curly brace of the lambda function.

3.17 Tableaux

Exemple :

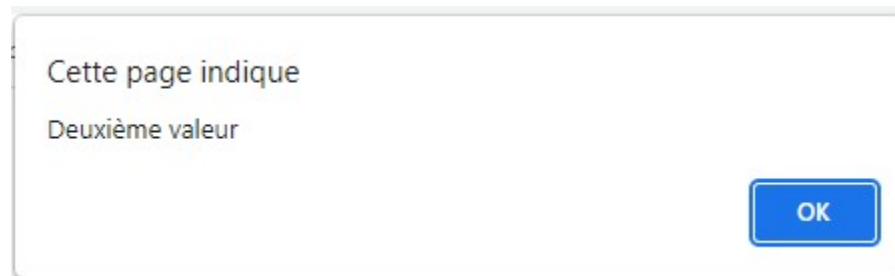
```
var myTable = new Array("Bonjour" , "Salut" , "Hello");  
  
myTable.forEach(element =>  
{  
    alert(element);  
});
```


3.18 Exercices

Exercice 16 :

Créer un tableau avec 3 valeurs. Afficher la deuxième valeur du tableau.
Commenter votre code. Respecter le camel case

Résultat attendu :



3.18 Exercices

Exercice 16 solution :

```
var myTable = new Array("Première valeur" , "Deuxième valeur" , "Troisième valeur");  
alert(myTable[1]);
```

3.18 Exercices

Exercice 17 :

Créer un tableau avec 5 valeurs numérique. Parcourir le tableau puis lorsque la valeur est égale à 9, afficher la position.

Commenter votre code. Respecter le camel case

Résultat attendu :



3.18 Exercices

Exercice 17 solution :

```
var myTable = new Array(12 , 8, 3, 56, 9, 32);  
for (var index = 0; index < myTable.length; index++) {  
    if (myTable[index] == 9) {  
        alert(`Le chiffre 9 a été trouvé en position ${index}`);  
    }  
}
```

3.18 Exercices

Exercice 18 :

Créer un tableau avec les jours de la semaine. Parcourir le tableau et afficher une phrase différente pour chaque jour rencontré.
Commenter votre code. Respecter le camel case.

Résultat attendu :

Cette page indique
Le lundi annonce une nouvelle semaine

OK

Cette page indique
Le mercredi signifie le milieu de semaine

OK

Cette page indique
Qui dit vendredi, dit week-end !

OK

Cette page indique
Le mardi montre que nous avons bien commencé la semaine

OK

Cette page indique
Le jeudi un est jour plutot cool

OK

Cette page indique
J'adore le samedi, je peux dormir

OK

Cette page indique
Dimanche c'est NETFLIX et repos !

OK

3.18 Exercices

Exercice 18 solution :

```
var days = new Array("Lundi" , "Mardi" , "Mercredi" , "Jeudi" , "Vendredi" , "Samedi" , "Dimanche");
days.forEach(element => {
  switch (element) {
    case "Lundi":
      alert("Le lundi annonce une nouvelle semaine");
      break;
    case "Mardi":
      alert("Le mardi montre que nous avons bien commencé la semaine");
      break;
    case "Mercredi":
      alert("Le mercredi signifie le milieu de semaine");
      break;
    case "Jeudi":
      alert("Le jeudi un est jour plutot cool");
      break;
    case "Vendredi":
      alert("Qui dit vendredi, dit week-end !");
      break;
    case "Samedi":
      alert("J'adore le samedi, je peux dormir");
      break;
    case "Dimanche":
      alert("Dimanche c'est NETFLIX et repos !");
      break;
  }
});
```

3.18 Exercices

Exercice 19 :

Demander la taille du tableau, puis le créer.

Demander ensuite la valeur de chaque ligne du tableau, puis afficher le résultat.

Commenter votre code. Respecter le camel case.

Résultat attendu :

Cette page indique

Donne moi la taille du tableau ?

OK Annuler

Cette page indique

Donne moi la valeur 0 du tableau ?

OK Annuler

Cette page indique

Donne moi la valeur 1 du tableau ?

OK Annuler

Cette page indique

Ligne : 0 | Valeur : 12

OK

Cette page indique

Ligne : 1 | Valeur : 34

OK

3.18 Exercices

Exercice 19 solution :

```
var size = prompt("Donne moi la taille du tableau ?");

// Création du tableau
var myTable = new Array(Number(size));

// Renseignement des valeurs
for (var index = 0; index < myTable.length; index++) {
    myTable[index] = prompt(`Donne moi la valeur ${index} du tableau ?`);
}

// Affichage des valeurs
for (var index = 0; index < myTable.length; index++) {
    alert(`Ligne : ${index} | Valeur : ${myTable[index]}`);
}
```


3.19 Recherche dans un tableau

Recherche d'un élément :

Recherche d'une valeur précise dans un tableau avec une condition lors du parcours de celui-ci.

Exemple :

```
var myTable = new Array(12 , 8, 3, 56, 9, 32);

for (var index = 0; index < myTable.length; index++)
{
    if (myTable[index] == 9)
    {
        alert(`Le chiffre 9 a été trouvé en position ${index}`);
    }
}
```

3.19 Recherche dans un tableau

Recherche d'un élément avec flag :

Recherche d'une valeur précise dans un tableau avec une condition lors du parcours de celui-ci.

Changement de valeur du flag lorsque la valeur a été trouvé.

Flag :

Variable contenant une valeur booléenne indiquant si un événement à eu lieu ou non.

3.19 Recherche dans un tableau

Exemple :

```
var myTable = new Array(12 , 8, 3, 56, 9, 32);
var flag = false;

for (var index = 0; index < myTable.length; index++)
{
    if (myTable[index] == 9)
    {
        flag = true;
    }
}

if (flag)
{
    alert("J'ai trouvé !");
}
```

3.19 Recherche dans un tableau

Sortir d'une recherche :

Lors d'un parcours d'un tableau il est possible de sortir de manière prématurée de la boucle (for, foreach, do/while, while) avec l'instruction **break** afin de ne pas parcourir tout le tableau inutilement.

Exemple :

```
var myTable = new Array(12 , 8, 3, 56, 9, 32);

for (var index = 0; index < myTable.length; index++)
{
    if (myTable[index] == 9)
    {
        alert(`J'ai trouvé | Position ${index}`);
        break;
    }
}
```

3.19 Recherche dans un tableau

Continuer une recherche :

Lors d'un parcours d'un tableau il est possible que nous souhaitons directement passer à l'élément suivant du tableau, l'instruction **continue** permet ceci.

Exemple :

```
var myTable = new Array(12 , 8, 3, 56, 9, 32, 9);

for (var index = 0; index < myTable.length; index++)
{
    if (myTable[index] == 9)
    {
        alert(`J'ai trouvé | Position ${index}`);
        continue;
    }
}
```

3.20 Recherche du plus petit / grand

Recherche du plus petit / grand élément :

Sont concernés : Les tableaux de valeur numérique

Méthodologie :

- Initialiser une variable avec une valeur du tableau
- Parcourir le tableau
- Tester si chaque valeur du tableau est plus petite/grande
 - Si une valeur est plus petite/grande, affecter la valeur du tableau à la variable

3.20 Recherche du plus petit / grand

Exemple :

Plus grande valeur :

```
var myTable = new Array(12 , 8, 3, 56, 9, 32, 9);

var intMax = Number(myTable[0]);

for (var index = 0; index < myTable.length; index++)
{
    if (myTable[index] > intMax)
    {
        intMax = myTable[index];
    }
}
```

Plus petite valeur :

```
var myTable = new Array(12 , 8, 3, 56, 9, 32, 9);

var intMin = Number(myTable[0]);

for (var index = 0; index < myTable.length; index++)
{
    if (myTable[index] < intMin)
    {
        intMin = myTable[index];
    }
}
```

3.21 Exercices

Exercice 20 :

Créer un tableau avec 3 valeurs numériques.

Parcourir le tableau puis effectuer une recherche de la valeur 5 avec un flag.

Commenter votre code. Respecter le camel case.

Résultat attendu :



3.21 Exercices

Exercice 20 solution :

```
var myTable = new Array(8 , 56, 5);
var flag = false;

for (var index = 0; index < myTable.length; index++) {
    if (myTable[index] == 5) {
        flag = true;
    }
}

// Condition sur la valeur booléenne
if (flag) {
    alert("J'ai trouvé la valeur 5 dans le tableau !");
} else {
    alert("Je n'ai pas trouvé !");
}
```

3.21 Exercices

Exercice 21 :

Demander la taille du tableau, puis le créer.

Demander ensuite la valeur (numérique) de chaque ligne du tableau.

Afficher le tableau et indiquer la valeur la plus grande.

Commenter votre code. Respecter le camel case.

Résultat attendu :

The image displays a sequence of five Java Swing dialog boxes, each with a title bar and a close button, illustrating the steps of a program to find the maximum value in an array.

- Dialog 1:** Title: "Cette page indique". Content: "Donne moi la taille du tableau ?". Input field: "2". Buttons: "OK" (blue), "Annuler" (grey).
- Dialog 2:** Title: "Cette page indique". Content: "Donne moi la valeur 0 du tableau ?". Input field: "12". Buttons: "OK" (blue), "Annuler" (grey).
- Dialog 3:** Title: "Cette page indique". Content: "Ligne : 0 | Valeur : 12". Button: "OK" (blue).
- Dialog 4:** Title: "Cette page indique". Content: "Donne moi la valeur 1 du tableau ?". Input field: "34". Buttons: "OK" (blue), "Annuler" (grey).
- Dialog 5:** Title: "Cette page indique". Content: "La valeur la plus grande du tableau est : 34". Button: "OK" (blue).

3.21 Exercices

Exercice 21 solution :

```
var size = prompt("Donne moi la taille du tableau ?");

// Création du tableau
var myTable = new Array(Number(size));

// Renseignement des valeurs
for (var index = 0; index < myTable.length; index++) {
    myTable[index] = prompt(`Donne moi la valeur ${index} du tableau ?`);
}

var intMax = Number(myTable[0]);

// Affichage des valeurs
for (var index = 0; index < myTable.length; index++) {
    alert(`Ligne : ${index} | Valeur : ${myTable[index]}`);
    if (myTable[index] > intMax) {
        intMax = myTable[index];
    }
}

alert(`La valeur la plus grande du tableau est : ${intMax}`);
```

3.21 Exercices

Exercice 22 :

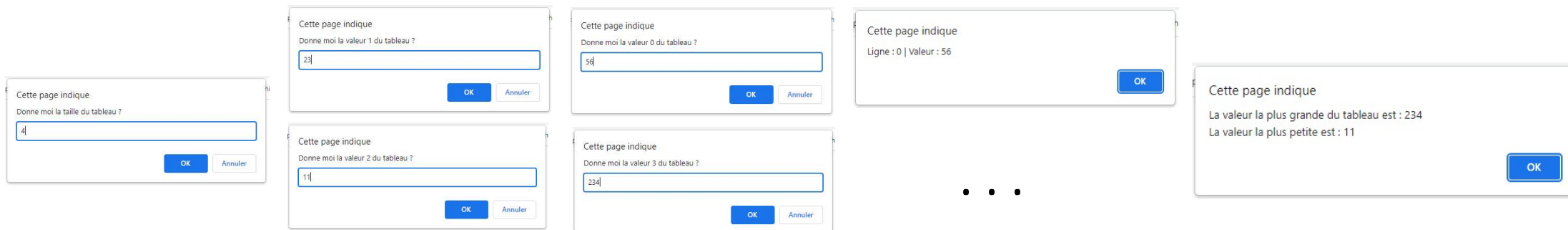
Demander la taille du tableau, puis le créer.

Demander ensuite la valeur (numérique) de chaque ligne du tableau.

Afficher le tableau et indiquer la valeur la plus grande ET la plus petite.

Commenter votre code. Respecter le camel case.

Résultat attendu :



The image shows a sequence of five dialog boxes illustrating the expected user interaction for Exercise 22:

- Dialog 1:** "Cette page indique" / "Donne moi la taille du tableau ?" with an input field containing "4" and "OK" / "Annuler" buttons.
- Dialog 2:** "Cette page indique" / "Donne moi la valeur 1 du tableau ?" with an input field containing "23" and "OK" / "Annuler" buttons.
- Dialog 3:** "Cette page indique" / "Donne moi la valeur 0 du tableau ?" with an input field containing "56" and "OK" / "Annuler" buttons.
- Dialog 4:** "Cette page indique" / "Donne moi la valeur 2 du tableau ?" with an input field containing "11" and "OK" / "Annuler" buttons.
- Dialog 5:** "Cette page indique" / "Donne moi la valeur 3 du tableau ?" with an input field containing "234" and "OK" / "Annuler" buttons.

Below these dialogs are three dots "...", indicating the sequence continues for all 4 elements.

Final Dialog: "Cette page indique" / "La valeur la plus grande du tableau est : 234" / "La valeur la plus petite est : 11" with an "OK" button.

3.21 Exercices

Exercice 22 solution :

```
var size = prompt("Donne moi la taille du tableau ?");

// Création du tableau
var myTable = new Array(Number(size));

// Renseignement des valeurs
for (var index = 0; index < myTable.length; index++) {
    myTable[index] = prompt(`Donne moi la valeur ${index} du tableau ?`);
}

var intMax = Number(myTable[0]);
var intMin = Number(myTable[0]);

// Affichage des valeurs
for (var index = 0; index < myTable.length; index++) {
    alert(`Ligne : ${index} | Valeur : ${myTable[index]}`);
    if (myTable[index] > intMax) {
        intMax = myTable[index];
    }

    if (myTable[index] < intMin) {
        intMin = myTable[index];
    }
}

alert(`La valeur la plus grande du tableau est : ${intMax} \nLa valeur la plus petite est : ${intMin}`);
```

3.21 Exercices

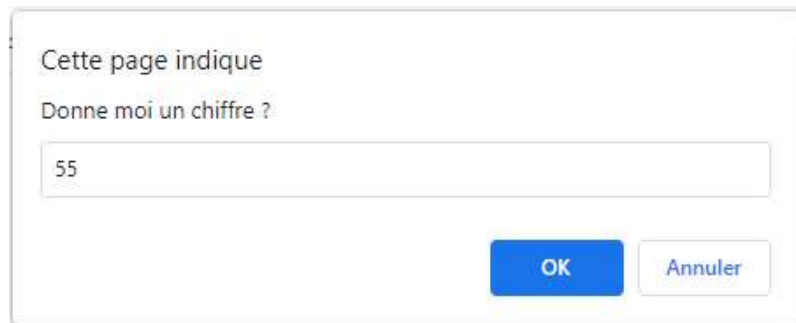
Exercice 23 :

Créer un tableau avec 10 valeurs numériques.

Demander un chiffre et indiquer combien de chiffres du tableau sont plus grands.

Commenter votre code. Respecter le camel case.

Résultat attendu :



Cette page indique

Donne moi un chiffre ?

55

OK Annuler



Cette page indique

Il y a 3 chiffre(s) dans le tableau qui sont plus grand

OK

3.21 Exercices

Exercice 23 solution :

```
var number = prompt("Donne moi un chiffre ?");
var myTable = new Array(8 , 56, 5, 67, 12, 98, 32, 5, 34, 44);
var count = 0;

for (var index = 0; index < myTable.length; index++) {
    if (myTable[index] > number) {
        // Compte les chiffres les plus grands
        count++;
    }
}

alert(`Il y a ${count} chiffre(s) dans le tableau qui sont plus grands`);
```

3.21 Exercices

Exercice 24 :

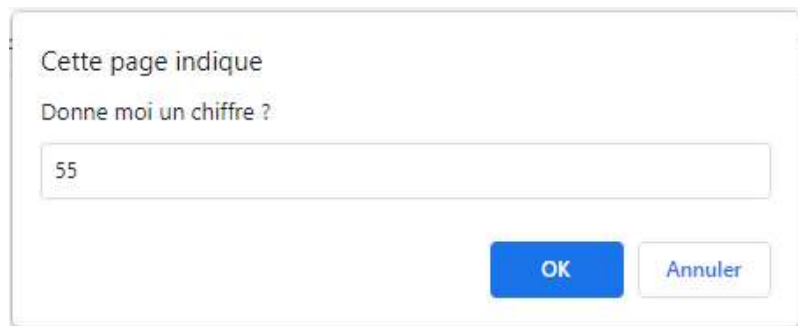
Créer un tableau avec 10 valeurs numériques.

Demander un chiffre et indiquer combien de chiffres du tableau sont plus grands.

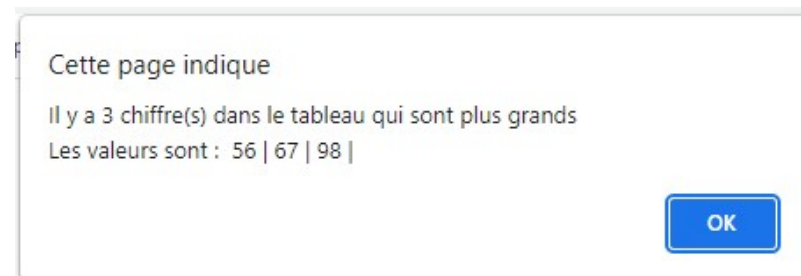
Afficher les valeurs plus grandes.

Commenter votre code. Respecter le camel case.

Résultat attendu :



A screenshot of a Java Swing dialog box titled "Cette page indique". The text inside says "Donne moi un chiffre ?". Below the text is a text input field containing the number "55". At the bottom right, there are two buttons: "OK" (highlighted with a yellow border) and "Annuler".



A screenshot of a Java Swing dialog box titled "Cette page indique". The text inside says "Il y a 3 chiffre(s) dans le tableau qui sont plus grands" and "Les valeurs sont : 56 | 67 | 98 |". At the bottom right, there is a single "OK" button.

3.21 Exercices

Exercice 24 solution :

```
var number = prompt("Donne moi un chiffre ?");
var myTable = new Array(8 , 56, 5, 67, 12, 98, 32, 5, 34, 44);
var tall = "";
var count = 0;

for (var index = 0; index < myTable.length; index++) {
    if (myTable[index] > number) {
        // Compte les chiffres les plus grands
        count++;
        tall += ` ${myTable[index]} | `
    }
}

alert(`Il y a ${count} chiffre(s) dans le tableau qui sont plus grands \nLes valeurs sont : ${tall}`);
```

3.22 Tri par insertion

Tri par insertion :

Retourne un tableau trié par ordre croissant ou décroissant

Méthodologie :

- Placer l'élément le plus petite dans la première case du tableau
- Mettre la deuxième plus petite valeur dans la deuxième case du tableau et ainsi de suite

3.22 Tri par insertion

Exemple :

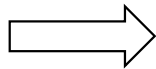
```
var myTable = new Array(8 , 56, 5, 67, 12, 98, 32, 5, 34, 44);
```

```
var min = 0;  
var tempo = 0;
```

```
for (var index = 0; index < myTable.length; index++) {  
    min = index;
```

```
    for (var indexMoreOne = index + 1; indexMoreOne < myTable.length; indexMoreOne++) {  
        if (myTable[indexMoreOne] < myTable[min]) {  
            min = indexMoreOne;  
        }  
    }  
}
```

Changement de signe
pour tri par insertion décroissant



```
    if (min != index) {  
        tempo = myTable[index];  
        myTable[index] = myTable[min];  
        myTable[min] = tempo;  
    }  
}
```

3.23 Tri à bulles

Tri à bulles :

Retourne un tableau trié par ordre croissant ou décroissant

Méthodologie :

- Parcourir le tableau
- Parcourir le tableau avec un décalage d'une valeur
- Chaque valeur est comparée avec celle qui le suit
- Si l'élément suivant est plus grand ou plus petit
 - Échange des deux valeurs
- Recommencer l'opération jusqu'à ne plus avoir d'échange possible

3.22 Tri à bulles

Exemple :

```
var myTable = new Array(8 , 56, 5, 67, 12, 98, 32, 5, 34, 44);
var tempo = 0;
for (var index = 0; index < myTable.length; index++)
{
    for (var indexMoreOne = index + 1; indexMoreOne < myTable.length; indexMoreOne++)
    {
        if (myTable[index] > myTable[indexMoreOne])
        {
            tempo = myTable[index];
            myTable[index] = myTable[indexMoreOne];
            myTable[indexMoreOne] = tempo;
        }
    }
}
```

Changement de signe
pour tri par insertion décroissant

