

IFT1015 Programmation 1

Énoncés conditionnels

if/else

Matériel produit par Marc Feeley et Pascal Vincent
(légèrement édité par Aaron Courville)

L'énoncé if



- L' énoncé **if** permet de faire une **exécution conditionnelle** basée sur une valeur booléenne
- Syntaxe : **if** (*<exp.>*) *énoncé*¹ *<énoncé>*
- Syntaxe : **if** (*<exp.>*) *énoncé*¹ **else** *<énoncé>*₂ 2
- L'évaluation de *<exp.>* **se fait en premier** 1
- Si la valeur est **true** l'exécution de *<énoncé>* suivra; si la valeur est **false** et qu'il y a un *énoncé*¹ **else**, c'est l'exécution de *<énoncé>*₂ qui suivra

L'énoncé if

- Exemple : obtenir deux nombres de l'utilisateur et afficher la valeur la plus grande

```
> var a = prompt("Nombre #1")  
> var b = prompt("Nombre #2")  
> alert(Math.max(+a, +b))
```

ou

```
> var a = prompt("Nombre #1")  
> var b = prompt("Nombre #2")  
> if (+a > +b) alert(a); else alert(b);
```

Exemple avec if

- Exemple : programme calculant la valeur absolue

```
// Fichier: abs.js

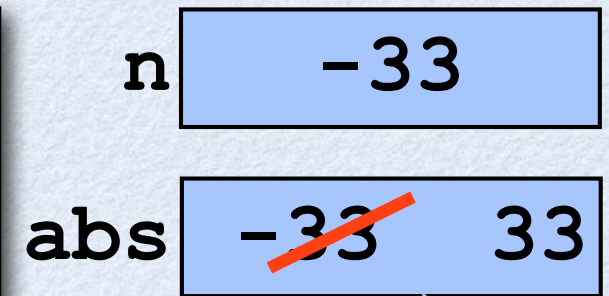
// Ce programme demande à l'utilisateur
// d'entrer un nombre et affiche à
// l'utilisateur la valeur absolue du nombre.

var n = +prompt("Nombre?");

var abs = n;

if (n < 0) // le nombre est-il négatif?
    abs = -n;

alert("La valeur absolue est " + abs);
```



La variable **abs**
n'est jamais
undefined

Exemple avec if/else

- Exemple : programme calculant la valeur absolue

```
// Fichier: abs2.js

// Ce programme demande à l'utilisateur
// d'entrer un nombre et affiche à
// l'utilisateur la valeur absolue du nombre.

var n = +prompt("Nombre?");

var abs;

if (n < 0) // le nombre est-il négatif?
    abs = -n;
else
    abs = n;

alert("La valeur absolue est " + abs);
```

n **-33**

abs ~~undefined~~ **33**

La variable **abs**
est **undefined**
au début

Blocs d'instructions

- On peut définir une **séquence** d'instructions à exécuter en l'englobant dans { }

- ```
if (<condition>)
 {
 instruction1;
 instruction2;
 ...
 }
else
 {
 instruction1;
 instruction2;
 ...
 }
```

## Indentation:

Dans un bloc, décalez toutes les instructions vers la droite.

- Indentation: faites commencer toutes les instructions

# Différents styles d'indentation et de placement des {}

## Style 1

```
if (<condition>)
{
 instruction1;
 instruction2;
 ...
}
else
{
 instruction1;
 instruction2;
 ...
}
```

bloc indenté →

bloc indenté →

## Style 2 (plus compact)

```
if (<condition>) {
 instruction1;
 instruction2;
 ...
} else {
 instruction1;
 instruction2;
 ...
}
```

le plus compact

# Ex avec blocs d'instructions

```
// Fichier: abs3.js

// Ce programme demande à l'utilisateur
// d'entrer un nombre et affiche à
// l'utilisateur la valeur absolue du nombre.

var n = +prompt("Nombre?");

var abs;

if (n < 0) // le nombre est-il négatif?
 { print("nombre négatif");
 abs = -n;
 }
else
 { print("nombre positif");
 abs = n;
 }
alert("La valeur absolue est " + abs);
```



# Attention!

- N'oubliez pas les {}, l'indentation ne suffit pas!

```
// Fichier: abs.js

// Ce programme demande à l'utilisateur
// d'entrer un nombre et affiche à
// l'utilisateur la valeur absolue du nombre.

var n = +prompt("Nombre?");

var abs = n;

if (n < 0) // le nombre est-il négatif?
 print("Nombre négatif");
 abs = -n;

alert("La valeur absolue est " + abs);
```

Programme  
buggé!

Que fera-t-il?

# Attention!

- Pas de ; après les () d'un if !

```
// Fichier: abs.js

// Ce programme demande à l'utilisateur
// d'entrer un nombre et affiche à
// l'utilisateur la valeur absolue du nombre.

var n = +prompt("Nombre?");
var abs = n;

if (n < 0); // le nombre est-il négatif?
{
 print("Nombre négatif");
 abs = -n;
}

alert("La valeur absolue est " + abs);
```

Programme  
buggé!

Que fera-t-il?

# Attention!

- Pas de ; après les () d'un if !

```
// Fichier: abs.js

// Ce programme demande à l'utilisateur
// d'entrer un nombre et affiche à
// l'utilisateur la valeur absolue du nombre.

var n = +prompt("Nombre?");
var abs = n;

if (n < 0) // le nombre est-il négatif?
{
 print("Nombre négatif");
 abs = -n;
}

alert("La valeur absolue est " + abs);
```

Programme  
buggé!

Que fera-t-il?

# Plusieurs alternatives avec else if

## Structure if imbriquées

```
var num = +prompt("Nombre?");
if (num==10){
 texte = "dix"; }
else {
 if (num==100) {
 texte = "cent";}
 else {
 if (num==1000){
 texte = "mille"; }
 else {
 texte = "ni dix, ni cent, ni mille";}}}
alert(texte);
```

# Plusieurs alternatives avec else if

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

```
var num = +prompt("Nombre?");
var texte;
if (num==10)
{
 texte = "dix";
}
else if (num==100)
{
 texte = "cent";
}
else if (num==1000)
{
 texte = "mille";
}
else
{
 texte = "ni dix, ni cent, ni mille";
}
alert(texte);
```

# switch

- Syntaxe:

```
switch (expression) {
 case label1:
 instruction1;
 instruction2;
 ...;
 break;
 case label2:
 instruction2;
 instruction3;
 ...;
 break;
 ...
 case labelN:
 instructionN;
 ...;
 break;
 default:
 instructions par défaut;
}
```

- L'expression est évaluée
- L'exécution saute au label qui a cette valeur
- Sinon elle saute à default:
- **break;** saute à la fin du switch
- Les `case label: ... break;` jouent le rôle des blocs `{ ... }` des `if`
- N'oubliez pas les **break!**

# switch

## if / else if / else

```
var num = +prompt("Nombre?");
var texte;

if (num==10)
{
 texte = "dix";
}
else if (num==100)
{
 texte = "cent";
}
else if (num==1000)
{
 texte = "mille";
}
else
{
 texte = "autre";
}


alert(texte);
```

## switch / case ... break / default

```
var num = +prompt("Nombre?");
var texte;

switch (num) ex: si num vaut 100
{
 case 10:
 texte = "dix";
 break;
 case 100:
 texte = "cent";
 break;
 case 1000:
 texte = "mille";
 break;
 default:
 texte = "autre";
}

alert(texte);
```



# switch: attention au break!

## ● Exemple 1:

```
switch (expression) {
 case label1:
 case label2:
 instruction1;
 break;
 default:
 instructions par défaut;
}
```

- L'énoncés ~~instruction1~~ est évaluée:

si expression == label1  
ou expression == label2

## ● Exemple 2: pas de break!

```
switch (expression) {
 case label1:
 instruction1;
 case label2: ←
 instruction2;
 break;
 default:
 instructions par défaut;
}
```

- L'énoncés ~~instruction1~~ est évaluées:

si expression == label1

- L'énoncés ~~instruction2~~ est évaluées:

si expression == label1  
ou expression == label2



# Expression conditionnelle

- L'énoncé **if** permet de choisir l'exécution d'un de deux énoncés en fonction d'une valeur booléenne
- Syntaxe : **if** (*<exp.>*) *<énoncé>* **else** *<énoncé>*
- L'expression conditionnelle est l'analogue pour choisir l'une de deux expressions :
- Syntaxe : *<exp1.>* ? *<exp2.>* : *<exp3.>*
- *<exp1.>* est évaluée, si vraie alors *<exp2.>* est évaluée, sinon *<exp3.>* est évaluée

**essai==1** ? "" : "s"