Fundamental notions of programming (NFP)

Class 3

Data

Data and data types

Data can be defined as information which can be used by a program. We, humans, can make a difference between a name, an address and a phone number. But, computers can't. Programming uses different codes allowing to establish the types of data being used.

String

A string is formed by characters (letters, numbers, symbols and spaces) used as text. They are placed between quotation marks.

```
print "1, 2, 3, Go! This is a string although there are numbers"
```

Numbers

There are two types of numbers: integers and floating points (floats).

Integers

Numbers used as values for operations coded without quotation marks.

```
print 15
```

Floating point (float or Double)

Numbers with decimals coded without quotation marks

```
print 3.1416
```

Boolean

Booleans are the two results of a condition's verification which can be ***true*** or ***false***.

```
if (age < 18) {
        // Under 18 (true): will print you are minor
}
        // Otherwise (false): will print you are of legal age
```

Arrays

An array consists in a list of data elements used to store multiple values of different types in a single variable. Their values are coded within straight brackets.

```
myArray = ["3 houses", 25, true, 3.14];
```

Variables

Variables are words used to store data (values) to be referenced and manipulated by the program. Think of it as containers holding informations and their main purpose is to label and store data in the computer's memory so they can be used by the program whenever needed.

Naming variables

Naming variables is an important task that shouldn't be taken lightly. Name your variables accurately so they perfectly describe the values they carry to anyone reading the program. Variables names should be as short as possible and they need to be understood by co-workers or remembered if you have to work on a program a few months or even years after it was created.

Bad :

```
nav1_0a          // that means nothing without reference
```

Good :

```
mainNav          // main navigation is easy to understand
```

Most of the time, but depending on the language, variables are case-sensitive, they can contain numbers but should start with a letter and must never contain spaces. Some programming languages forbid the use of certain words reserved for predefined values. Finally, if a variable contains many words, it is considered best practice to use camel case.

Camel case :
Camel case consists in writing sentences so each word begins with a capital letter (except the first one).

Example :
thisVariableIsReallyTooLong

Declaration of variables

Declaring variable consists into telling a program a certain word will be used as a variable. Depending on the language, the declaration will differ. For instance, the words « **var** » or « **let** » will be used in JavaScript while PHP will use the dollar sign (**$**).

JavaScript :

```
var myVariable;
let myVariable;
```

PHP :

```
$myVariable;
```

Variable assignment

Variable assignment consists into setting or re-setting the value contained in a variable. To do so, the equal (**=**) operator is usually used to associate the variable and its value which can be of any type.

JavaScript :

```
var userFirstName;
let userAge;

userFirstName = "John";
userAge = 23;

Or more simply :
let userFirstName = "John";
```

PHP :

```
$interestRate = 2.99;
```

NOTE :

After a variable has been declared, the value may be modified simply by using the equal (=) sign to associate the variable to its new value.

```
JavaScript :
userAge = 33
```

Difference between data and variables

Data are constants (language specific predefined variables) or fixed informations the program uses. Variables, on their parts, may vary. The values they originally contain may change throughout the course of the program.

Mathematic operations

Mathematic operations are quite easy using any programming languages and values as well as results can be assigned to variables which can also be used in different operations.

Arithmetic operators

Operator	Description
+	**Addition :** Adds values. Used to get the sum of numbers or to concatenate strings.
-	**Subtraction :** Subtracts the right hand operand from the left hand operand to get the difference.
*	**Multiplication :** Multiplies values (factors). Used to get the product.
/	**Division :** Divides the left hand operand (dividend) by the right hand operand (divisor). Used to get the quotient.
%	**Modulo (remainder) :** Divides the left hand operand by the right hand operand and returns remainder.

Order of operations

The order of operations is the same in programming than for any mathematic operations. This means divisions and multiplications must be calculated first, and always starting with operations within parenthesis.

2 + 3 * 4 = 14

Note :
*3 * 4 must be calculated first, then proceed with the addition.*

4 * (7 + 6) / 4 = 13

Note :
(7 + 6) must be calculated first, then proceed from left to right with the multiplication and the division.

Using variables

When using variables in an instruction, to output to screen, for instance, they are coded without quotation marks, like in the following example *(the examples syntax and structure are willingly wrong)* :

Example outputting to screen :

```
$userFirstName = "John";
$userLastName = "Smith";
$userAge = 23;

print $userFirstName;          // John
print $userLastName;           // Smith
print $userAge;                // 23
```

Using variables with numbers

Just like it is done with strings, numbers can be assigned to variables so these can be used in mathematical operations.

```
let a = 2;
let b = 3;
let c ;
c = a + b                      // c equals 5
```

Variable can also be used in conditions and be compared

Example in a condition :

```
$userAge = 23;
$legalAge = 18;
IF ($userAge < 18);
        print "You are under 18";

  OR
  IF ($userAge < $legalAge);
          print "You are under 18";

ELSE
        print "You are of legal age";
```

Concatenation of variables

It is possible to assemble variables together so they would be output together on the same line, for instance. To do so, Javascript uses the plus (**+**) sign and PHP uses a period (**.**).

Concatenating strings and integer :

```
$userFirstName = "John";
$userLastName = "Smith";
$userAge = 23;
print $userFirstName . $userLastName . " : " . $userAge;
```

Result : John Smith : 23

Scope of variables

Variables can be global or local. When a variable is global, it means it can be used anywhere within the program. On the contrary, local scope variables can only be used within a fragment of the program, a function, for instance.

```
let myName = "John";                    // Global
print myName;

function myFunction {
        Let myCity = "Montreal";        // Local
        print MyName;
        print myCity;
}
print myCity;                           // this wouldn't print
```

Explanation :
The variable ***myName*** is global and can be used anywhere in the program. Having been declared within a function, the variable ***myCity*** is local and can only be used within the function into which it was declared.

A bit of practice

JavaScript *promt()* command

Prompt() is a command that pops up an alert window allowing users to fill in a field. The answer submitted by the user can be assigned to a variable so it can be used in many ways.

In an HTML document, in the <body> section, write the following codes :

```
<script>
prompt("What is your first name?");
</script>
```

Explanation :
prompt() pops an alert window containing a field preceded by a string (the question). But if a user writes a name in the field, the data is lost since it wasn't stored anywhere. It must be assigned to a variable so we can use it.

```
<script>
let firstName = prompt("What is your first name?");
document.write(firstName);
</script>
```

Explanation :
The variable *firstName* is declared and assigned the *promt()* command. Then, using *document.write()*, the variable's value is output to screen.

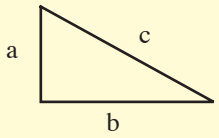
```
<script>
let firstName = prompt("What is your first name?");
let lastName = prompt("What is your last name?");
document.write(firstName + " " + lastName);
</script>
```

Explanation :
The variables *firstName* and *lastName* are declared and assigned different *promt()* commands so the user is asked two different questions one after the other. Then, using *document.write()*, the variables values are concatenated using the «+» signs (also with a string to create a space between the two values) and output to screen altogether on one line.

Assignment 3 : Pythagorean theorem

Create the pseudocode and flowchart, using variables with the correct data types and scopes, then write the following program :

The user is asked to enter the length of sides *a* and *b* of a triangle.



A function is created calculates the hypotenuse (side c) using the formula : $c = \sqrt{(a^2 + b^2)}$, or more simply : a2 + b2 = c2.

Finally, the result has to be output to screen like in the following example :

Example :
The hypotenuse of you triangle is : *c variable's value*.