Scripting language I (LS1)

Course 9

Programming workshop

Programming of numerical to text number converter

The program to create will ask the user to write a number in an input field and will return the text version of the number. For instance, if the user enters « 123 », the message displayed in the message window will be : « One hundred and twenty-three ».

Program flow :

- The user is asked to write a number between 0 and 999.
- The number is sent to a function which converts it in word(s).
- This function will have to decompose the number in units, tens and hundreds. It then needs to be able to identify units, tens and hundreds of a given number.
- When the number will have been decomposed in 3 different numbers, these will have to be converted into words and then displayed.
- When the returned text converted number is display, the user is asked to choose another number.

Preamble : Test and number conversion

parseInt :

We all normally make mathematic operation on base 10. But other bases could be used and some browsers uses base 8 or 16, so it is important to insure base 10 will be used. . Nous calculons normalement sur une *base de 10*, mais il est possible de faire autrement et certains fureteurs n'utilisent pas la base de 10 par défaut.

On base 10, the following instruction should display 10 :

```
alert(parseInt("010"));
```

We will use *parseInt* to convert the string in number, but to specify to use base 10, we will add it as second value to the object :

```
alert(parseInt("010", 10));
```

isNaN() :

isNaN(), meaning *is not a number*, is a function returning a Booleans to identify if a value is a number or a string. For a string, the returned value will be *true*, and the returned value for a number will be *fals*.

```
alert(isNaN(test));          // Displays : « true »
alert(isNaN(128));           // Displays : « false »
```

Step 1: Initializing the conversion function

It is first needed to ask the user for a number and to create the conversion function. The argument *number* and the variables must be verified so the program works correctly.

Argument's verification:

The first part of the following source code declare a function named *converter*. The argument named *number* is used in the function's parenthesis and refers to the user's chosen number.

The function first verifies if the user input is a numerical value using isNaN(). If it's not a numerical value, an error message is returned and the user is asked to choose a number again.

The second part of the source code (out of the function) declares the variable in which the user data will be stored (*user*) using *prompt()* within a conditional loop. If user stops supplying numbers, the function stops.

The function is finally called within an alert window and the user inout data is converted from string to number in base 10 using *ParseInt()*.

```
function converter(number) {
        if (isNaN(number) || number < 0 || 999 < number) {
                return "Please choose a number between 0 and 999.";
        }
}

/* USER INPUT */
let user;
while (user = prompt("Choose a number between 0 and 999 :")) {
        alert(converter(parseInt(user, 10)));
}
```

Step 2: Declaring the arrays

In order to display the text version of the number chosen by the user, we will need to use words we can select to form the final spelled number: the text version of all the different numbers divided in hundreds, tens and units stored in three arrays.

```
let     unitsTxt = ["", "one", "two", "three", "four", "five", "six", "sev
        en", "eight", "nine", "ten", "eleven", "twelve", "thirthteen", "four
        teen", "fifteen", "sixteen", "seventeen", "eighteen", "nineteen"],

        tensTxt = ["", "", "twenty", "thirty", "forty", "fifty", "sixty",
        "seventy", "eighty", "ninety"],

        hundredsTxt = ["", "one", "two", "three", "four", "five", "six",
        "seven", "eight", "nine"];
```

Step 3: Extracting hundreds, tens and units

In order to associate the numbers to the correct array and index, we have to divide and extract the hundreds, tens and units from the user's number using the modulo operator and store them into variables.

Working with modulo:

If we take the following modulo operation, **365 % 10**, the result consists in the **remaining of the division of the number** by 10, i.e. **5** (the units).

If we take the following modulo operation, **365 % 100**, the result is **65**, we then subtract units (65 - 5 = 60), and we divide by ten to obtain 6 (tens).

For the hundred, the principle is the same, but the units and tens are subtracted.

```
let     units = number % 10,
        tens = (number % 100 - units) / 10,
        hundreds = (number % 1000 - number % 100) / 100;

let     unitsOut, tensOut, hundredsOut;
```

Step 4: Creating an exception for zero and the units output

If the user chosen number is 0, then the returned value will be « zero », otherwise, we store the text version of the units in a variable (*unitsOut*). To do so, we access the units array (*unitsTxt*) and use the variable *units* to retrieve the proper index. Note that the first value of the arrays are empty since numbering starts with 0. This way, we avoid the usual shift.

```
if (number === 0) {
        return "zero";
        } else {
                unitsOut = unitsTxt[units];
        }
```

Step 5: Creating the hundreds output

Here, we need to associate the hundred numerical value to the according index of the hundreds array. There are two choices. Whether there are no hundreds or it is equal or higher than 1. Also, the text version of the hundreds must be followed by the word « hundred » or replaced by nothing if there are no hundreds.

```
if (hundreds >= 1) {
        hundredsOut = hundredsTxt[hundreds] + " hundred ";
} else if (hundreds < 1) {
        hundredsOut = hundredsTxt[hundreds] + "";
}
```

Step 6: Creating the units and tens ending with the suffix « teen »

Although, because there are numbers such as oneteen, twoteen, etc., we have to add 10 to the index.

```
if (tens === 1 && units >= 0) {
        tensOut = tensTxt[0];
        unitsOut = unitsTxt[10 + units];
        } else if (1 <= units <= 9) {
                tensOut = tensTxt[0];
                unitsOut = unitsTxt[units];
        }
```

Step 7: Creating the tens (2 and over)

Here, we proceed the same way, but let's not forget a dash must be used between the tens and the units and those ending with 0 are one word only (no units).

```
if (tens > 1 && units === 0 ) {
        tensOut = tensTxt[tens];
        unitsOut = unitsTxt[units];
} else if (tens > 1) {
        tensOut = tensTxt[tens] + "-";
        unitsOut = unitsTxt[units];
}
```

Step 8 (final): Returning the string

Finally, we are ready to return the final values using the output variables created.

```
return hundredsOut + tensOut + unitsOut;
```

Complete code

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
</head>
<body>
<script>
/*  FUNCTION DECLARATION  */

function converter(number) {
	if (isNaN(number) || number < 0 || 999 < number) {
		return "Please choose a number between 0 and 999.";
	}

/* ARRAYS AND VARIABLES DECLARATION AND INITIALIZATION */

let	unitsTxt = ["", "one", "two", "three", "four", "five", "six", "seven", "eight", "nine", "ten", "eleven", "twelve",
	"thirthteen", "fourteen", "fifteen", "sixteen", "seventeen", "eighteen", "nineteen"],

	tensTxt = ["", "", "twenty", "thirty", "forty", "fifty", "sixty", "seventy", "eighty", "ninety"],

	hundredsTxt = ["", "one", "two", "three", "four", "five", "six", "seven", "eight", "nine"];

let	units = number % 10,
	tens = (number % 100 - units) / 10,
	hundreds = (number % 1000 - number % 100) / 100;

let	unitsOut, tensOut, hundredsOut;

/* ZERO */

if (number === 0) {
		return "zero";
	} else {
		unitsOut = unitsTxt[units];
	}

/* HUNDREDS */

if (hundreds >= 1) {
	hundredsOut = hundredsTxt[hundreds] + " hundred ";
} else if (hundreds < 1) {
	hundredsOut = hundredsTxt[hundreds] + "";
}
```

```
/* UNITS AND TEENS */

        if (tens === 1 && units >= 0) {
                tensOut = tensTxt[0];
                unitsOut = unitsTxt[10 + units];
        } else if (1 <= units <= 9) {
                tensOut = tensTxt[0];
                unitsOut = unitsTxt[units];
        }

/* TENS */

if (tens > 1 && units === 0 ) {

                tensOut = tensTxt[tens];
                unitsOut = unitsTxt[units];
        } else if (tens > 1) {

                tensOut = tensTxt[tens] + "-";
                unitsOut = unitsTxt[units];
        }

        return hundredsOut + tensOut + unitsOut;
}

/* USER INPUT */
        let user;
        while (user = prompt("Choose a number between 0 and 999 :")) {
                alert(converter(parseInt(user, 10)));
        }

</script>
</body>
</html>
```