Information and Communication Technologies — S7 — Work n°5

# PHP: VARIABLE MANIPULATION

Don’t forget: for now, you can test simple PHP code on:

https://phpsandbox.io/

https://www.w3schools.com/php/phptryit.asp?filename=tryphp_compiler

https://sandbox.onlinephpfunctions.com/

## 1 Strings

1. In PHP, you have two types of strings: with single quotes (’) and with double quotes ("). Let’s see the difference on some examples. Try to type the different lines in Listing 1, and see what happens.

   Bonus: try to modify the lines that do not work without changing the quotes (keep single quotes if they were used, keep double quotes if they were used), e.g. by using string concatenation as seen last week.

```
$nb_students = 7;
echo ’Our group contains $nb_students students’;
echo "Our group contains $nb_students students";
echo ’These apples cost $2 per kilogram’;
echo "These apples cost $2 per kilogram";
echo "Today’s our lucky day";
echo ’Today’s our lucky day’;
echo "This is what we call "bad luck"";
echo ’This is what we call "bad luck"’;
echo ’This week\’s been our "honey moon"’;
```

Listing 1: PHP strings

## 2 Transtyping

2. Do you remember what happens if you try Listing 2 in Python? How to make it work?

3. Try the equivalent in PHP, see Listing 3.

```
pi = "3.1415927"
radius = 5
print(pi * (radius * radius))
```

Listing 2: Python transtyping

```
$pi = "3.1415927";
$radius = 5;
echo $pi * ($radius * $radius);
```

Listing 3: PHP transtyping

# 3 Variable range

When using functions, a common mistake is to try to use variables that are not passed to the function, see Listing 4. In order to use the variable `$n` inside the function `expo`, you have (at least) three options, test them:

1. Add the line `global $n;` somewhere in the function.

2. Add the line `global $n;` when you first use it, outside the function.

3. Add a second argument to the function `expo`, and pass the `$n` variable as second argument when calling the function.

```
$n = 3;
function expo($x) {
    $a = 1;
    for($i = 0; $i < $n; $i++) {
        $a = $a * $x;
    }
    return $a;
}
echo expo(2);
```

Listing 4: PHP variable range

# 4 Some programming

4. Write a PHP function that takes a string as input, and checks whether there is an `@` in its characters. If yes, print “you entered a correct email address”, else print “you forgot a @ in your email address.”

   Hint: you should use the `strpos()` function, see documentation on https://www.php.net/strpos.

   Bonus: try to make this function smarter, e.g., detect other kind of strings which cannot be email addresses.

5. Write a PHP function that takes an integer `len` as input, and returns a string containing random characters (from the set of characters of your choice, e.g. only numbers or only letters), with length `len`.

   Hint: you can start with an empty string, and make a loop of size `len`, adding each time a random character in the current string. To choose a character to add to the string, you might want to use a specific string, e.g. `$possible_chars = "abcdef";`, and choose a random character in this string. To choose a random character in this string, you might want to generate a random number (here, in $\{0, 1, 2, 3, 4, 5\}$) by using the `mt_rand()` function, see documentation on https://www.php.net/mt_rand, and access the corresponding character in `$possible_chars`.

   Bonus: use this function to generate a random Morse code.