

## Worksheet 1: *Basic Hugs*

---

Template file:	Worksheet1.hs
Labs :	1/2/19
Hand-in:	10/2/18 at 18.00.
Topics:	Type synonyms. Pairs. Functions via patterns + Lists

---

Put your name and username on your submission.

Use the handin-system to handin

Make sure that your submission compiles in GHCi.

Try to give a nice layout to your answers. Particularly bad or unreadable layout may cost you points.

Each assessed question counts for 10 points.

1. Define a function `pastTense` that gives the past tense of most verbs in English, i.e. it adds an "ed" at the end of the word.
2. (ASSESSED) Define a function `bmi` to calculate "body mass index" from a given `mass` in kg and `height` in meters.  
Wikipedia [https://en.wikipedia.org/wiki/Body\\_mass\\_index](https://en.wikipedia.org/wiki/Body_mass_index) explains nicely the history behind this idea of body mass index, and provides the details of the function.
3. (ASSESSED) The daily costs incurred by a car plant is a function `cost` that depends on the number `n` of cars made per day. If  $0 \leq n$  and  $n \leq 500$  then the cost is  $5 * n + 1000$ . Otherwise the cost is  $10 * n + 450$ . Write a Haskell function coding this `cost` function.
4. Write a function that given a year of birth of a person (assuming that the person was born not more than 99 years ago...) given as two digits and calculates today's age of that person.
5. Today 1 British Pound Sterling equals 1.14 Euro. Write a function that converts a price in pounds into euros. And also a function which does the converse.  
(Euro sign can be typed using "Alt Gr" key together with 4)
6. (ASSESSED) Write two functions `prettyprintEuro` and `prettyprintPound` that take as input respectively an element of the above types Euro and Pound and produce as output a string as suggested by the examples: €5.00 and £5.00.

To test try `putStr (prettyprintEuro 5.00)` and `putStr (prettyprintEuro 5.00)`

If you are up for a (non-assessed) challenge, try and figure out how you can force haskell to print the first two decimals of a float...

7. (ASSESSED) Read the section on Escaping text on <http://book.realworldhaskell.org/read/characters-strings-and-escaping-rules.html>.

Now write down a string `rawtext` such that `putStr rawtext` outputs exactly

```
"This is a \ \long string,
\ \ spanning multiple lines,
in fact 3 lines!"
```

8. (ASSESSED) Using `filter` write a function `removeZeroes` that removes all zeroes from a given list of integers.
9. (ASSESSED) Using `map` write a function `capslockon` that replaces all the lowercase characters in a given string by the corresponding uppercase and all the uppercase characters by the corresponding lowercase, while leaving the other symbols unchanged. Example: `capslockon "Fer-Jan de Vries" = "fER-jAN DE vRIES"`
10. Challenge: investigate experimentally how many characters there are in `Char`, and write down a term `listOffAllCharacters` such that `putStr (show listOffAllCharacters)` prints all characters in their natural order.
11. Challenge: write a function `removeZeroes2` that given any non-zero integer `m` returns an integer with all zero digits that occur `m`, and produces an error message when the input is zero.

```
removeZeroes2 10120300 = 1123
```

Don't use functions from the web. You learn from doing it yourself! The idea is to convert a integer in a string, remove the 0 characters from the string and convert the string back into an integer. Hint: try to use recursion on lists...