Chapter 2- First Step

→ GHC (Glasgow Haskell Complier) is the leading developer for using Haskell

→ GHCi is an interpreter just like the Python, can ask it to evaluate simple math expressions

→ Haskell comes with a lot of standard library functions also includes useful functions on lists.

1. head [1,2,3,4,5] → Returns first element
   returns 1

2. tail [1,2,3, 4, 5] → Removes first element
   returns [2,3,4,5]

3. $\overset{0}{1}$, $\overset{1}{2}$, $\overset{2}{3}$
   [1 ,2, 3, 4, 5] !! 2
   ↳ Return the specified position element
   returns 3

4. take 3 [1,2, 3,4,5]
   ↳ # of elements (integer) to return from the start
   returns [1,2, 3]

5. drop 3 [1,2, 3,4,5]
   ↳ # of elements (integer) to remove from the start
   returns [4,5]

6. length [1,2, 3,4,5] → # of elements in list
   returns 5

7. sum [1, 2,3,4,5] → Addition of all the elements in list
   returns 15

8. product [1, 2, 3, 4, 5] → Multiplication of all the elements in list
returns 120

9. [1, 2, 3] ++ [4, 5] → Appends two lists
returns [1, 2, 3, 4, 5]

10. reverse [1, 2, 3, 4, 5] → Reverses the list
returns [5, 4, 3, 2, 1]

→ Function application is denoted using space
Ex: f a b + c * d → In Math: f(a, b) + c · d
→ Function application has higher priority than all operators
Ex: f a + b → Haskell uses f(a) + b

Cheat Sheet →

Math	Haskell
f(x)	f x
f(x, y)	f x y
f(g(x))	f (g x)
f(x, g(y))	f x (g y)
f(x) g(y)	f x * g y

→ Less amount of brackets

→ .hs is the file extension. Haskell script has many function definitions

→ GHCi does not automatically detect that script has been changed so have to perform reload command.

:reload
:load name → Loads the script
:type expr → Any Haskell expression

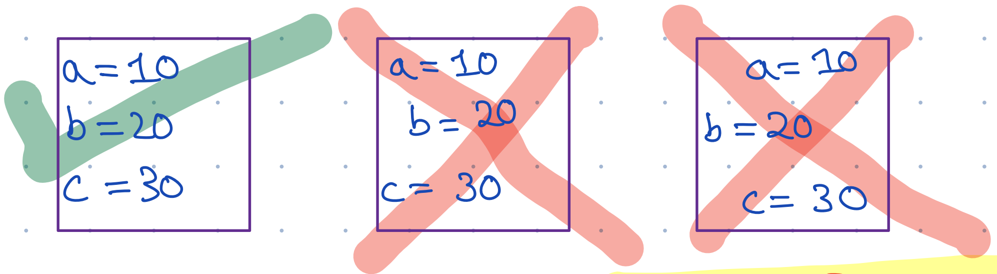
→ Can use one letter abbrevation

→Function names and argument names must begin with lowercase Ex: myFun arg_2

→ List arguments usually end with an s.

Ex: xs, ns, nss

→ Layout rule used by Haskell each definition must begin in same column.



```
a=10
b=20
c=30
```

```
  a=10
   b=20
c=30
```

```
  a=10
b=20
  c=30
```

Note Exercises for this lecture: firstScript.hs

→Let complier handle the efficient part of the code, focus on clear code.