

CSI31 Lecture 6

Topics:

- 3.1 Numeric Data Types
- 3.2 Using the Math Library
- 3.3 Accumulating Results: Factorial
- 3.4 The limits of Int
- 3.5 Handling Large Numbers: Long Ints
- 3.6 Type Conversions

3.1 Numeric Data Types

When computers were first developed, they were seen primarily as number crunchers.

data - is the information that is stored and manipulated by computer programs

Different kinds of data will be stored and manipulated in different ways.

Integers: ..., -3, -2, -1, 0, 1, 2, 3, ...

integer data type (int)

Decimals (numbers with fractional part): 0.123, -4.345

floating point data type (float)

try to input the following commands in Python interpreter, interactive window:

```
>>> type(23)
```

```
>>> type(1.23)
```

```
>>> type(2.0)
```

```
>>> my_number=23.1
```

```
>>> type(my_number)
```

The function `type` returns us the type of the value.

! float type stores only approximations to real numbers (there is a *limit to the precision*, or *accuracy*)

more of the Python built-in numeric operations:

`**` exponentiation, $2^{**}3 = 2^3 = 8$

`%` remainder, $12 \% 5 = 2$

`abs()` absolute value, $\text{abs}(-23.4) = 23.4$

try to input the following statements in the Python Interpreter:

```
>>> 2.3*9.8
```

```
>>> 2/4
```

```
>>> 2/4.0
```

```
>>>2.0/4
```

```
>>>2.0/4.0
```

3.2 Using the Math Library

Python provides many other useful mathematical operations in a special *math library*

A *library* - is just a module that contains some useful definitions (of functions).

In order to use functions from the library we need to *include it* or *import it* to our program:

```
import math
```

see table 3.2 on page 57 for some math library functions.

see also Python Documentation -> Library Reference -> Numeric and Mathematical Modules -> math

input the following statements in the Python Interpreter:

```
>>> import math
>>> math.sqrt(5)
```

```
>>> math.sqrt(25)
>>> math.ceil(234.345)
```

3.3 Accumulating Results: Factorial

factorial function: $n!$

$$n! = n(n-1)(n-2)(n-3) \dots * 3 * 2 * 1$$

examples: $2! = 2 * 1 = 2$

$4! = 4 * 3 * 2 * 1 = 12 * 2 = 24$

Let's write a program that calculates the factorial of a number entered by the user:

Input: a positive integer (n)

Output: a positive integer (factorial)

Relationship: $factorial = n(n-1)(n-2) \dots * 2 * 1$

Algorithm:

input number to take the factorial of, n

for loop that will iterate n times

 factorial = factorial * factor

output *factorial*

see the program: factorial.py