Expressions

- **Expressions** are a fundamental building block in programs

- Expressions are analogous to the idea of clauses in English
  - Single clause sentence:
    "I am a student."
  - Multiple clause sentence:
    "I am a student and I am currently in COMP110."
  - In English, *prose* is *more expressive* through the creative use of *clauses*

- In code, ***programs*** are *more expressive* through creative uses of ***expressions!***

Expressions

There are two **big ideas** behind expressions:

1. *Every* expression ***evaluates*** *to a single value at runtime*
   - Thus, every expression has a *single type*.
   - This occurs *only* when the program runs (runtime) and when the processor *evaluates* the expression.

2. Anywhere you can write an expression you can substitute any other expression *of the same type*

Expressions of Various Kinds

- Literal Values
  - `110`
  - `3.14`
  - `True`
  - `"hi"`

- Variable Access
  - `x`
  - `comp_course_number`

- "Unary" operators
  - `-x (number` *`negation)`*
  - `not True (boolean negation)`

- Function Calls
  - `abs(x) - absolute value of x`

- "Binary" Operators
  - Arithmetic
    - `1 + 2`
  - Concatenation
    - `"Hello " + name`
  - Equality
    - `x == 1`
    - `x != 1`
  - Relational
    - `age >= 21`
    - `age < 13`

Use *type casting* expressions to convert types

- When you have a value of one type but need to use it as another type

- A *type casting expression* can help you convert between types:
  `[desired type]([expression of starting type])`

- Examples:
  - **`str(110)`** evaluates to the string value **`"110"`**
  - **`str(100 + 10)`** evaluates to the string value **`"110"`**
  - **`int("3000")`** evaluates to the integer value **`3000`**
  - **`float("3.14")`** evaluates to the floating-point value **`3.14`**
  - **`int(3.99)`** evaluates to the integer **`3`**
    - float to integer conversions truncate, or get rid of, the decimal component "rounds down"

- Warning: if a value cannot be cast to the desired type a ValueError will result.
  - Try: `int("hello, world")`

- In Python, a *type casting expression* is a special *function call* we'll discuss soon!