

Outline for January 10, 2018

Reading: §2, 3

1. Input
 - a. `input`, `raw_input`
 - b. `try ... except`
2. Simultaneous assignment [*swap.py*]
 - a. Simple assignment: `variable = expression`
 - b. Simultaneous assignment: `variableA, variableB = expressionA, expressionB`
3. Decision structures [*if0.py*]
 - a. If statement
 - b. Executes once, based on condition
 - c. Syntax
4. Conditions
 - a. Resolves to boolean value
 - b. Literal booleans: True (1), False (0)
 - c. Testable as `true` or `false`
 - d. Relational operators
 - i. Use two arithmetic expressions connected with relational operators to create a boolean
 - ii. Relational operators: `>`, `>=`, `<`, `<=`, `==`, `!=`
 - iii. Precedence: resolved after arithmetic operators
 - iv. `6 > 2 + 3`; `"UCD" == "Sac State"`
5. Two-way decisions [*if1.py*]
 - a. `if ... else` statements
 - b. One condition, two possible code blocks
 - c. Syntax
 - d. `else` very powerful when the positive condition is easy to describe but not the negative
 - e. String comparison example
6. Multi-way decisions [*if2.py*]
 - a. Can execute code based on several conditions
 - b. `elif` (else if)
 - c. Syntax
 - d. `else` only reached if all previous conditions false
 - e. Nested if statements