

Making Decisions In Python

In this section of notes you will learn how to have your Pascal programs choose between alternative courses of action.

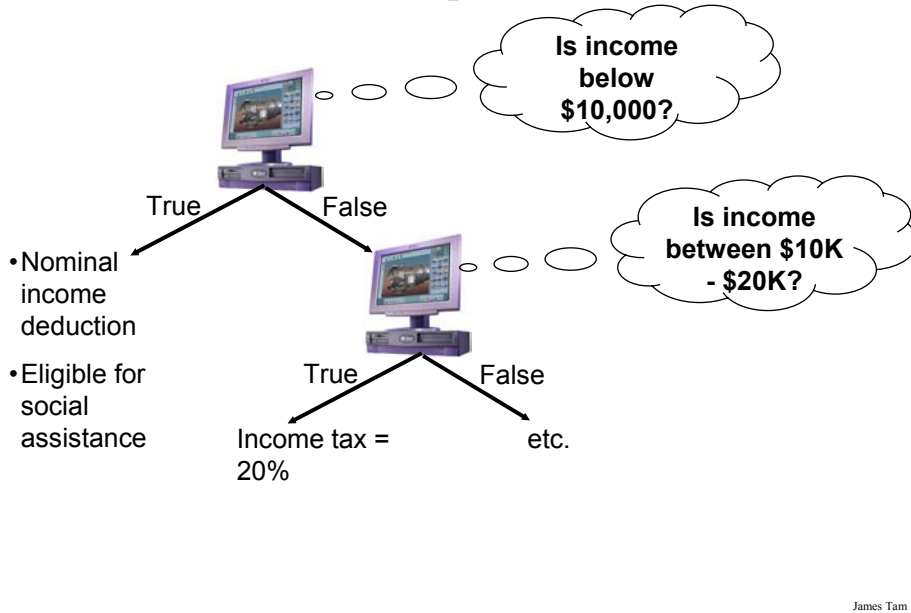
James Tam

Why Is Branching/Decision Making Needed?

- When alternative courses of action are possible and each action may result in a different result.
- Branching/decision making can be used in a program to deal with alternative.

James Tam

High Level View Of Decision Making For The Computer



Decision-Making In Python

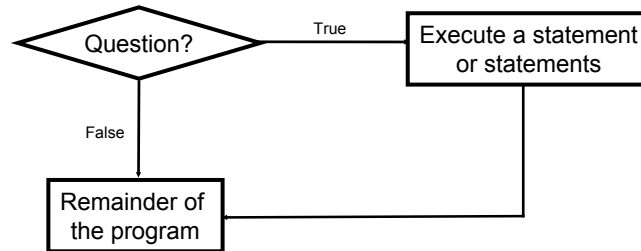
Decisions are questions with answers that are either true or false (Boolean) e.g., Is it true that the variable 'num' is positive?

The program branches one way or another depending upon the answer to the question (the result of the Boolean expression).

Decision making/branching constructs (mechanisms) in Python:

- If
- If-else
- If-elif-else

Decision Making With An 'If'



James Tam

The 'If' Construct

Decision making: checking if a condition is true (in which case something should be done).

Format:

(General format)

```
if (Boolean expression):  
    body
```

(Specific structure)

```
if (operand relational operator operand):  
    body
```

Boolean expression

Note: Indenting the body is mandatory!

James Tam

The 'If' Construct (2)

Example:

```
if (age >= 18):  
    print "You are an adult"
```

James Tam

Allowable Operands For Boolean Expressions

If (operand relational operator operand) then:

Some operands

- integer
- real numbers
- String

Make sure that you are comparing operands of the same type!

James Tam

Allowable Relational Operators For Boolean Expressions

If (operand relational operator operand) then

Python operator	Mathematical equivalent	Meaning	Example
<	<	Less than	5 < 3
>	>	Greater than	5 > 3
==	=	Equal to	5 == 3
<=	≤	Less than or equal to	5 <= 5
>=	≥	Greater than or equal to	5 >= 4
<>	≠	Not equal to	5 <> 5
OR			
!=			5 != 5

James Tam

If (Simple Body)

Body of the if consists of a single statement

Format:

Indenting used to indicate
what statement is the body

```
if (Boolean expression):
```

```
    s1
```

Body

```
s2
```

Example:

```
if (num == 1):
```

```
    print "Body of the if"
```

```
print "After body"
```

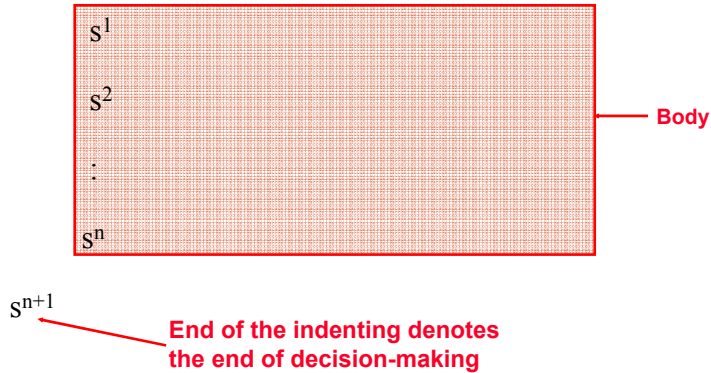
James Tam

If (Compound Body)

Body of the if consists of multiple statements

Format:

if (Boolean expression):



James Tam

If (Compound Body(2))

Example:

```
taxRate = 0.2;  
if (income < 10000):  
    print "Eligible for social assistance"  
    taxCredit = 100  
tax = (income * taxRate) - taxCredit
```

James Tam

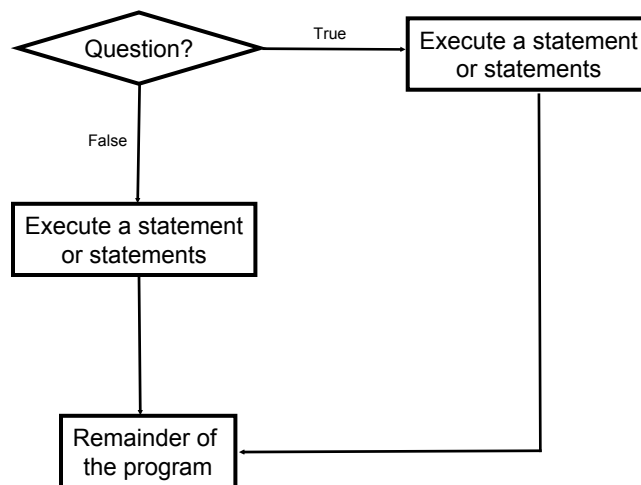
Decision Making With An 'If': Summary

Used when a question (Boolean expression) evaluates only to a true or false value (Boolean):

- If the question evaluates to true then the program reacts differently. It will execute a body after which it proceeds to execute the remainder of the program (which follows the if construct).
- If the question evaluates to false then the program doesn't react different. It just executes the remainder of the program (which follows the if construct).

James Tam

Decision Making With An 'If-Else'



James Tam

The If-Else Construct

Decision making: checking if a condition is true (in which case something should be done) but also reacting if the condition is not true (false).

Format:

```
if (operand relational operator operand):
```

```
    body of 'if'
```

```
else:
```

```
    body of 'else'
```

```
additional statements
```

James Tam

If-Else Construct (2)

Example:

```
if (age >= 18):
```

```
    print "Adult"
```

```
else:
```

```
    print "Not an adult"
```

```
print "Tell me more about yourself"
```

James Tam

If-Else (Compound Body(2))

Example:

```
if (income < 10000):  
    print "Eligible for social assistance"  
  
    taxCredit = 100  
  
    taxRate = 0.1  
else:  
    print "Not eligible for social assistance"  
  
    taxRate = 0.2  
tax := (income * taxRate) - taxCredit;
```

James Tam

Quick Summary: If Vs. If-Else

If:

- Evaluate a Boolean expression (ask a question)
- If the expression evaluates to true then execute the 'body' of the if.
- No additional action is taken when the expression evaluates to false.
- Use when your program evaluates a Boolean expression and code will be executed only when the expression evaluates to true.

If-Else:

- Evaluate a Boolean expression (ask a question)
- If the expression evaluates to true then execute the 'body' of the if.
- If the expression evaluates to false then execute the 'body' of the else.
- Use when your program evaluates a Boolean expression and different code will execute if the expression evaluates to true than if the expression evaluates to false.

James Tam

Decision-Making With Multiple Expressions

Format:

```
if (Boolean expression) logical operator (Boolean expression):  
    body
```

Example:

```
if (x > 0) and (y > 0):  
    print "X is positive, Y is positive"
```

James Tam

Decision-Making With Multiple Expressions (2)

Commonly used logical operators in Python

- or
- and
- not

James Tam

Forming Compound Boolean Expressions With The “OR” Operator

Format:

```
if (Boolean expression) or (Boolean expression):  
    body
```

Example:

```
if (gpa > 3.7) or (yearsJobExperience > 5):  
    print “You are hired”
```

James Tam

Forming Compound Boolean Expressions With The “AND” Operator

Format:

```
if (Boolean expression) and (Boolean expression):  
    body
```

Example:

```
if (yearsOnJob <= 2) and (isGoofOff = True):  
    print “You are fired”
```

James Tam

Forming Compound Boolean Expressions With The “NOT” Operator

Format:

```
if NOT (Boolean expression):  
    body
```

Examples:

```
if NOT ((x > 0) and (y > 0)):  
    print “NAND”  
  
if NOT ((x > 0) or (y > 0)):  
    print “NOR”  
  
if NOT (x == 0):  
    print “X is anything but zero”
```

James Tam

Quick Summary: Using Multiple Expressions

Use multiple expressions when multiple questions must be asked and the result of each expression may have an effect on the other expressions:

AND:

- All Boolean expressions must evaluate to true before the entire expression is true.
- If any expression is false then whole expression evaluates to false

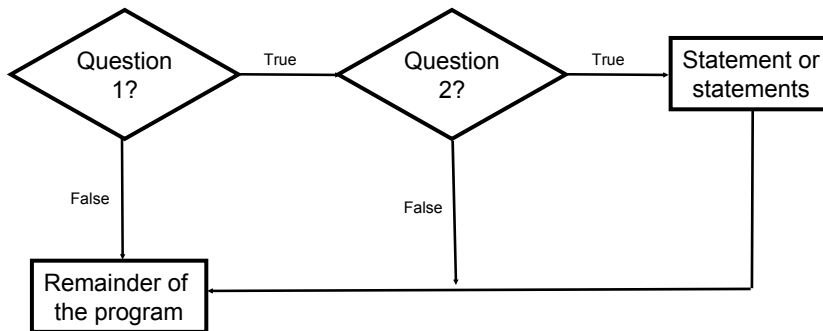
OR:

- If any Boolean expression evaluates to true then the entire expression evaluates to true.
- All Boolean expressions must evaluate to false before the entire expression is false.

James Tam

Nested Decision Making

- Decision making is dependent.
- The first decision must evaluate to true before successive decisions are even considered for evaluation.



James Tam

Nested Decision Making

One decision is made inside another.

Outer decisions must evaluate to true before inner decisions are even considered for evaluation.

Format:

if (Boolean expression):

if (Boolean expression):

inner body

Outer body

Inner body

James Tam

Nested Decision Making (2)

Example:

```
if (income < 10000):  
    if (citizen == 'y'):  
        print "This person can receive social assistance"  
        taxCredit = 100  
tax = (income * TAX_RATE) - taxCredit
```

James Tam

Decision-Making With Multiple Alternatives

if

Checks a condition and executes the body of code if the condition is true

if-else

Checks a condition and executes one body of code if the condition is true and another body if the condition is false

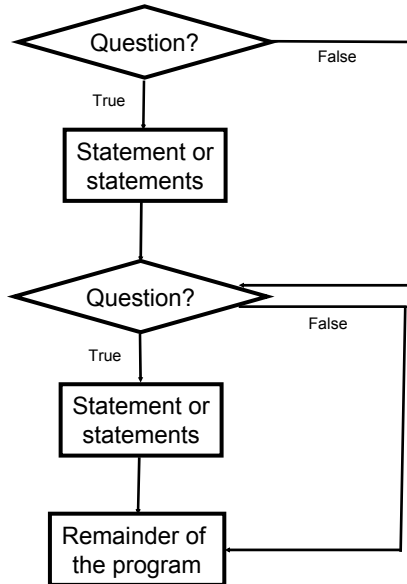
Approaches for multiple (two or more) alternatives

Multiple if's

if-elif-else

James Tam

Decision Making With Multiple If's



James Tam

Multiple If's: Non-Exclusive Conditions

Any, all or none of the conditions may be true (independent)

Format:

if (Boolean expression 1):

body 1

if (Boolean expression 2):

body 2

:

statements after the conditions

James Tam

Multiple If's: Non-Exclusive Conditions (Example)

Example:

```
if (num1 > 0):  
    print "num1 is positive"  
  
if (num2 > 0):  
    print "num2 is positive"  
  
if (num3 > 0):  
    print "num3 is positive"
```

James Tam

Multiple If's: Mutually Exclusive Conditions

At most *only one* of many conditions can be true

Can be implemented through multiple if's

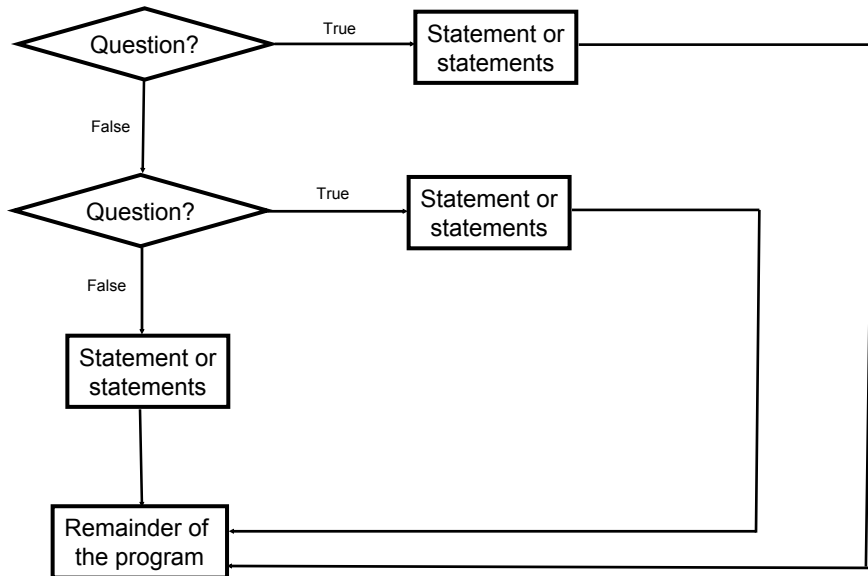
**Inefficient
combination!**

Example (for full example look in Unix under
`/home/courses/217/examples/decisions/inefficient.py`)

```
if (gpa == 4):  
    letter = 'A'  
  
if (gpa == 3):  
    letter = 'B'  
  
if (gpa == 2):  
    letter = 'C'  
  
if (gpa == 1):  
    letter = 'D'  
  
if (gpa == 0):  
    letter = 'F'
```

James Tam

Decision Making With If-Elif-Else



James Tam

Multiple If-Elif-Else: Mutually Exclusive Conditions

Format:

```
if (Boolean expression 1):  
    body 1  
elif (Boolean expression 2):  
    body 2  
    :  
else  
    body n  
statements after the conditions
```

James Tam

Multiple If, Else-If's: Mutually Exclusive Conditions (Example)

Example (the full version can be found in UNIX under /home/courses/217/examples/decisions/efficient.py):

```
if (gpa == 4):  
    letter = 'A'  
  
elif (gpa == 3):  
    letter = 'B'  
  
elif (gpa == 2):  
    letter = 'C';  
  
elif (gpa == 1):  
    letter = 'D'  
  
elif (gpa == 0):  
    letter = 'F'  
  
else:  
    print "GPA must be one of '4', '3', '2', '1' or '1'"
```

This approach is more efficient when at most only one condition can be true.

The body of the else executes only when all the Boolean expressions are false. (Useful for error checking).

James Tam

Recap: What Decision Making Constructs Are Available In Pascal/When To Use Them

Construct	When To Use
If	Evaluate a Boolean expression and execute some code (body) if it's true
If-else	Evaluate a Boolean expression and execute some code (first body) if it's true, execute alternate code (second body) if it's false
Multiple if's	Multiple Boolean expressions need to be evaluated with the answer for each expression being independent of the answers for the others (non-exclusive). Separate code (bodies) can be executed for each expression.
If-elif-else	Multiple Boolean expressions need to be evaluated but zero or at most only one of them can be true (mutually exclusive). Zero bodies (if) or exactly one body will execute. Also it allows for a separate body (else) to execute when all the if-elif Boolean expressions are false.

James Tam

Recap: When To Use Compound And Nested Decision Making Constructs (2)

Construct	When To Use
Compound decision making	More than one Boolean expression must be evaluated before some code (body) can execute.
Nested decision making	The outer Boolean expression must be true before the inner expression will even be evaluated.

James Tam

Testing Decision Making Constructs

Make sure that the body of each decision making construct executes when it should.

Test:

- 1) Obvious true cases
- 2) Obvious false cases
- 3) Boundary cases

James Tam

Testing Decisions: An Example

```
num = input("Type in a value for num: ")  
if (num >= 0):  
    print "Num is non-negative."  
else:  
    print "Num is negative."
```

James Tam

Avoid Using Real Values When An Integer Will Do

```
num = 1.0 - 0.55  
if (num == 0.45):  
    print "Forty five"  
else:  
    print "Not forty five"
```

James Tam

You Should Now Know

What are the three decision making constructs available in Python:

- If
- If-else
- If-elif-else
- How does each one work
- When should each one be used

How to evaluate and use decision making constructs:

- Tracing the execution of simple decision making constructs
- How to evaluate nested and compound decision making constructs and when to use them

James Tam

You Should Now Know (2)

How the bodies of the decision making construct are defined:

- What is the body of decision making construct
- What is the difference between decision making constructs with simple bodies and those with compound bodies

What is an operand

What is a relational operator

What is a Boolean expression

How multiple expressions are evaluated and how the different logical operators work

How to test decision making constructs

James Tam