Improving our first program

JAN 30, 2015

Making the program more robust

- What if the user types in a negative integer or 0?

  Or a real number? Or some non-numeric string, (e.g., "hello")?

- We will only discuss the negative integer or 0 situation now.

- Later when we discuss *exceptions* and how to handle them, we'll return to this program.

Types of errors

- *Syntax* error

  Syntax refers to the structure or form of the program.

  (e.g., English sentences start with a capital letter)

  Examples:

```
while x < 10
        x =x + 1
```

```
n = int(input()
print(n)
```

Types of errors

- *Run-time* errors (or *exceptions*)

  This is an error that occurs during the running of the program and is typically caused by the user not anticipating a certain behavior of their program.

  Example:

```
n = int(input("Enter a number:"))
print(n + 5)
```

What if the user inputs "hello"?

Types of errors

- *Semantic* errors

  The program may not produce an error message when executed, but it may not do what we expect it to do.

  Example:

  In an earlier version of our program:

```
print("The binary equivalent of", n, "is", suffix)
```

  We forgot that n would have changed to 0 at this point.

The case of negative integers

- What does the program currently do, if the user inputs a negative integer or 0?
- We could instead try to print an informative message if input is negative.
- We will use the `if-else` statement for that.

Simple if statement

```
Line 1
if  boolean expression:
        Line 2
        Line 3
Line 4
```

- Possibility 1:

  Line 1, bool expr (True), Line 2, Line 3, Line 4.
- Possibility 2:

  Line 1, bool expr (False) Line 4.

if-else statement

```
Line 1
if boolean expression:
        Line 2
        Line 3
else:
        Line 4
Line 5
```

- Possibility 1:
  Line 1, bool expr (True), Line 2, Line 3, Line 5

- Possibility 2:
  Line 1, bool expr (False), Line 4, Line 5

Dealing with negative input

One possible approach:

- If n < 0, print out an appropriate message and do nothing else.
- Else, continue to do what the program is currently doing.

Note: We still have to deal with the n equals 0 case.

Our Final First Program

```
n = int(input("Type a nonnegative integer: "))

if n < 0:
    print("Please input a nonnegative integer next time. Bye!")
else:
    originalN = n
    suffix = ""
    while n > 0:
        suffix = str(n % 2) + suffix
        n = n//2

    # The input n = 0 is dealt with as a special case
    if suffix == "":
        suffix = "0"

    print("The binary equivalent of", originalN, "is", suffix)
```