
VARIABLES AND ASSIGNMENT

Python Programming — Auburn University

BASIC ASSIGNMENT

- Variables do not need to be declared unless you need to alter their scope.
 - We will talk more about scope later.
 - Assignment binds a value to a variable (creating the variable if needed):

```
a = "monkey"
banana = 33
tree = 5*banana
```
 - Variables do not have types, values do.
-

CASCADED ASSIGNMENT

- Works as expected:
 $a = b = 99$
assigns 99 to both a and b

WALRUS OPERATOR

- Python supports “assignment as an expression” using the walrus operator `:`
 - Expression vs statement
 - normal assignment is a statement
 - walrus assignment is an expression (has a value)
 - Useful, for example, when assigning during loop iteration:

```
while (line := f.readline()) != "":  
    do_something(line)
```
 - <https://www.python.org/dev/peps/pep-0572/#exceptional-cases> — ugh
-

VARIABLE NAMING CONVENTIONS

- <https://www.python.org/dev/peps/pep-0008/#naming-conventions>
 - Local variable names should be all lowercase letters with words separated by the underscore character:
 - Python: `number_of_antelopes`
 - Java: `numberOfAntelopes`
-

STRING LITERALS WITH VARIABLES

- Convenience syntax for creating strings that include expression value(s).
- Use `f''` and place expression in `{ }`:

```
a = 22
print(f"The value of a is {a}")
message = f"If I subtract 1 I get {a-1}"
```

- If the expression's value is not a string, it will be converted to one using `str()`.
 - Format specifiers can control width, justification and, for floats, precision of output. Demo.
 - See PEP-498 (<https://www.python.org/dev/peps/pep-0498/>)
-

