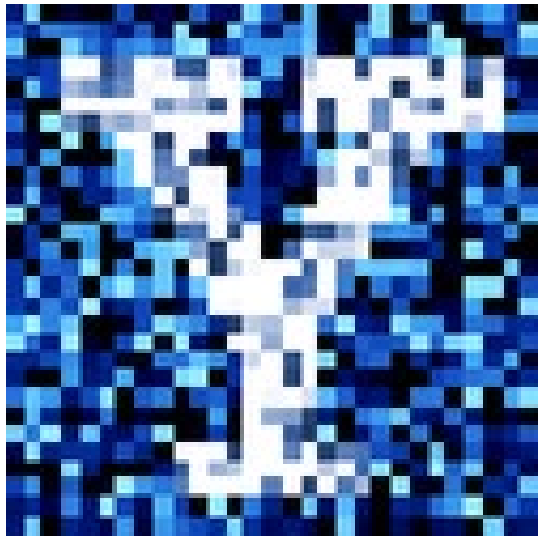
YData: Introduction to Data Science



Lecture 04: Data Types

Overview

Python continued!

Review and continuation of

- Functions
- Arithmetic operations

Strings

Types

Arrays

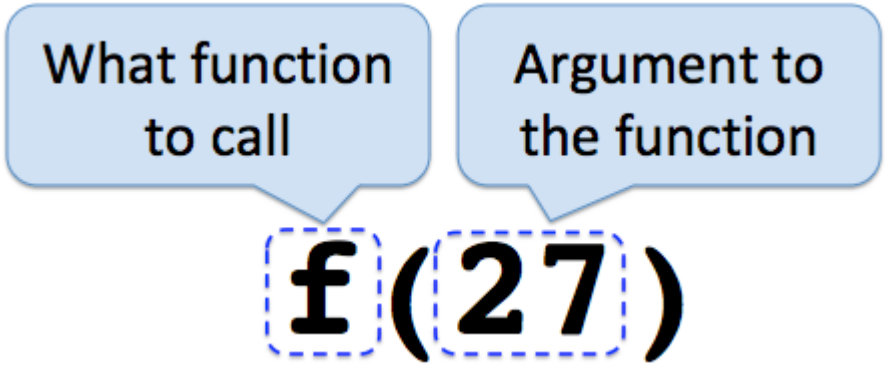
Tables (if there is time)

Review and continuation of functions

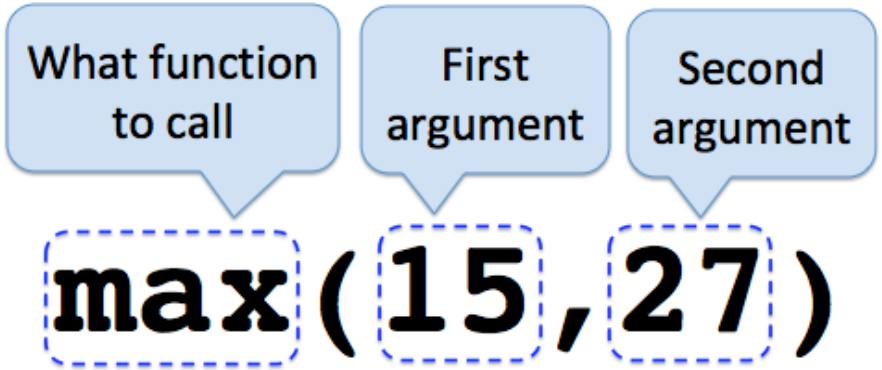
Let's pick up where we left off last class: Call Expressions

Call expressions are expressions that call functions

- Functions take in one or more values (arguments) and (usually) return another value



Example: taking the maximum value



Let's explore this in Jupyter!

Review and continuation of arithmetic

Arithmetic operations

Operation	Operation	Example	**Value**
Addition	+	2 + 3	5
Subtraction	-	2 – 3	-1
Multiplication	*	2*3	6
Division	/	7/3	2.667
Remainder	%	7 % 3	1
Exponentiation	**	2**.05	1.414

We can store the output of evaluating expression in names

- my_result = 10 * 2

Numbers in Python: Ints and Floats

Python has two real number types

- **int**: an integer of any size
- **float**: a number with an optional decimal part

An int never has a decimal point - a float always does

- 3 # int of float?
- 2.7 # int of float?

A float might be printed using scientific notation

Notes on Floats

Three limitations of float values:

- They have limited size (but the limit is huge)
- They have limited precision of 15 - 16 decimal places
- After arithmetic, the final few decimal places can be wrong

Let's explore this in Jupyter!

Arithmetic Question

What numbers are these expressions equal to?

A. 3 * 10 ** 10	A. 30000000000
B. 10 * 3 ** 10	B. 590490
C. (10 * 3) ** 10	C. 590490000000000
D. 10 / 3 / 10	D. 0.3333333333333337
E. 10 / (3 / 10)	E.33.333333333333336

Best to err on the side of using parentheses!

Strings

Text and Strings

A string value is a snippet of text of any length

- 'a'
- 'word'
- "there can be 2 sentences. Here's the second!"

Strings consisting of numbers can be converted to numbers

- int('12')
- float('1.2')

Any value can be converted to a string

- str(5)

Let's explore this in Jupyter!

Discussion Questions

Assume you have run the following statements

- x = 3
- y = '4'
- z = '5.6'

What's the source of the error in each example?

- A. x + y
- B. x + int(y + z)
- C. str(x) + int(y)
- D. str(x, y) + z

Types

Every value has a type

We've seen several types so far:

- int: 2
- Built-in function: abs()
- float: 2.2
- str: 'Red fish, blue fish'

The type function can tell you the type of a value

- type(2)
- type('Red fish')

An expression's type is based on its value, not how it looks

- x = 2
- type(x)

Let's explore this in Jupyter!

Conversions

Strings that contain numbers can be converted to numbers

- int('12')
- float('1.2')
- float('one point two') # Not a good idea!

Any numeric value can be converted to a string

- str(5)

Numbers can be converted to other numeric types

- float(1)
- int(1.2) # DANGER: loses information!

Arrays

Arrays (i.e., NumPy ndarrays)

An array contains a sequence of values

- All elements of an array must have the same type
- We can apply fast operations to all elements of an array
  - E.g., we can add a number to all elements of a numeric array
- When two arrays are added corresponding elements are added in the result
  - Note, the two arrays must have the same size

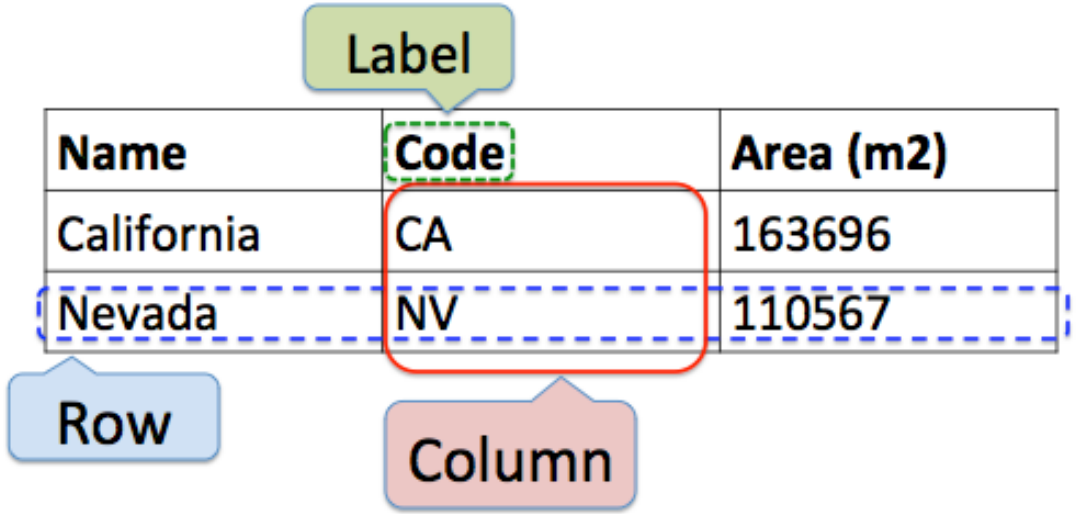
Let's explore this in Jupyter!

Tables

Table structure

A Table is a sequence of labeled columns

- Each row represents one individual case
- Data within a column represents one attribute



Some Table Operations

t.select(label) - constructs a new table with just the specified columns

t.drop(label) - constructs a new table in which the specified columns are omitted

t.sort(label) - constructs a new table with rows sorted by the specied column

t.where(label, condition) - constructs a new table with just the rows that match the condition

Let's explore this in Jupyter!

Discussion question

How to display just the row corresponding to the player who had the highest salary?

nba table

PLAYER	POSITION	TEAM	SALARY
Paul Millsap	PF	Atlanta Hawks	18.6717
Al Horford	C	Atlanta Hawks	12
Tiago Splitter	C	Atlanta Hawks	9.75625
Jeff Teague	PG	Atlanta Hawks	8
Kyle Korver	SG	Atlanta Hawks	5.74648
Thabo Sefolosha	SF	Atlanta Hawks	4
Mike Scott	PF	Atlanta Hawks	3.33333
Kent Bazemore	SF	Atlanta Hawks	2
Dennis Schroder	PG	Atlanta Hawks	1.7634
Tim Hardaway Jr.	SG	Atlanta Hawks	1.30452

Pandas

FYI: The datascience package is a Berkeley product

It's a light wrapper on top of pandas

Hopefully at the end of the class we'll have time to discuss Pandas

Review

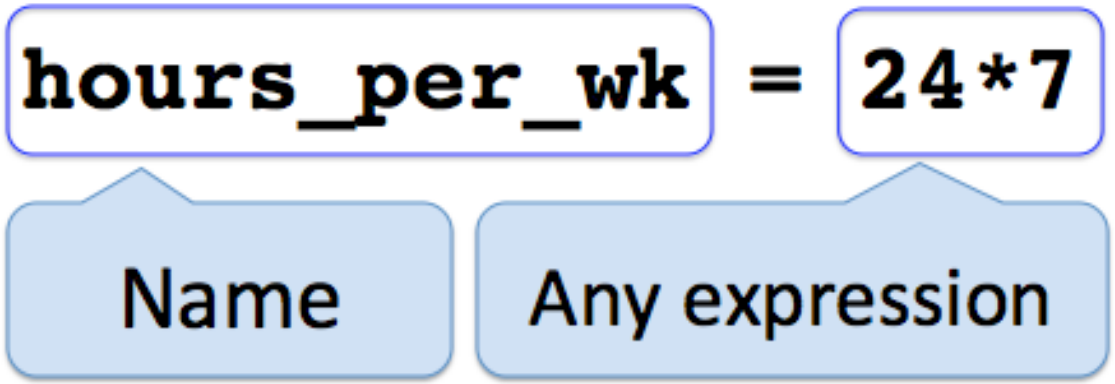
```
x = cones.select(`Flavor', `Color')

x

y = x.drop(`Color')
y

x = cones.select(`Color', `Price')
x
y
```

What are the column labels of each table?

Flavor	Color	Price
strawberry	pink	3.55
chocolate	light brown	4.75
chocolate	dark brown	5.25
strawberry	pink	5.25
chocolate	dark brown	5.25
bubblegum	pink	4.75

Demo

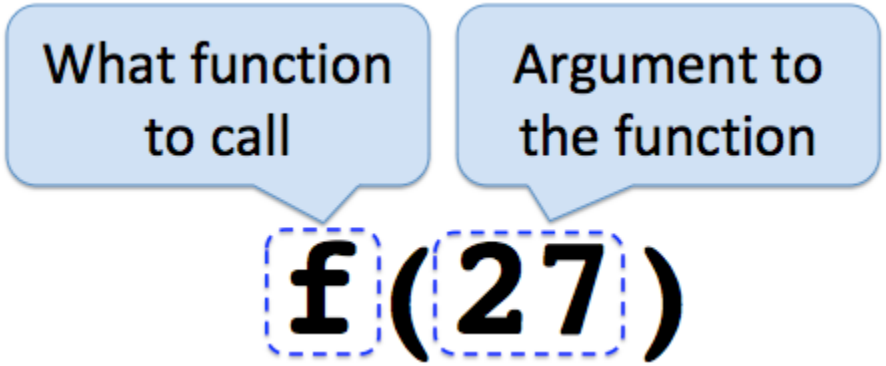
Assignment statements



- Statements don't have a value; they perform an action
- An assignment statement changes the meaning of the name to the left of the = symbol
- The name is bound to a value (not an equation)
- (DEMO)

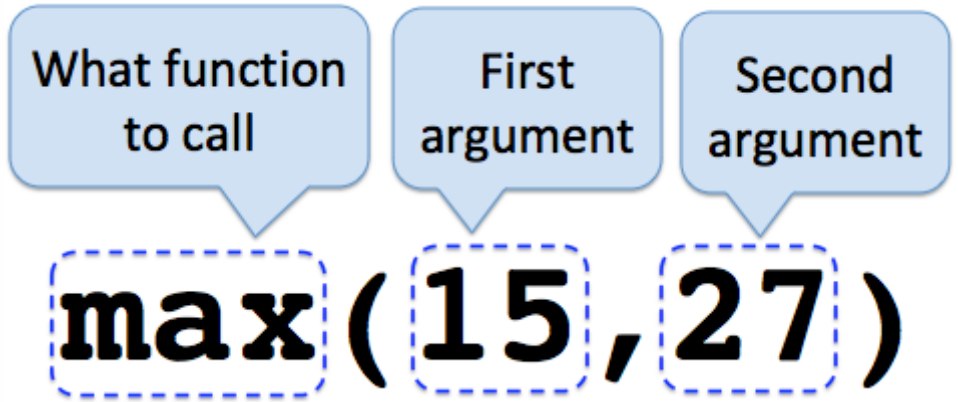
Call Expressions

Anatomy of a Call Expression



"Call f on 27"

Anatomy of a Call Expression



- Demo

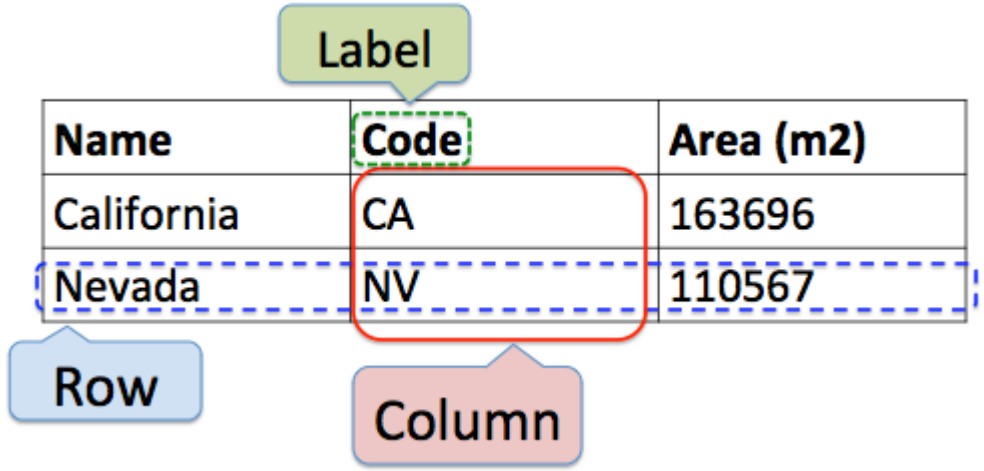
Tables

Table structure

- A Table is a sequence of labeled columns
- Each row represents one individual
- Data within a column represents one attribute of the individuals

- Did I go over this in the first two classes?

- demo



Some Table Operations

t.select(label) - constructs a new table with just the specified columns

t.drop(label) - constructs a new table in which the specified columns are omitted

t.sort(label) - constructs a new table with rows sorted by the specied column

t.where(label, condition) - constructs a new table with just the rows that match the condition

Discussion question

- nba table:
- How to display just the row corresponding to the player who had the highest salary?

Pandas

- FYI: The datascience package is a Berkeley product
- It's a light wrapper on top of pandas
- Later in the course we'll give an introduction to Pandas

Summary

Today we talked about how to:

- Assign a value to a name
- Call a function
- Build a Table (aka dataframe)
- Operate on Tables