

CS 114 Tutorial 9

Nov 14 2025

Goals for this Week:

- ❑ Have assignment 04 submitted (due next tonight)
- ❑ Look at assignment 05 (releases tonight)
- ❑ Understand how to use where when masking
- ❑ Know how to save and load an array using ssv
- ❑ Be familiar with classes

Building a Class

```
class StatsList:
    """
    Stores a list of number and provides some simple
    statistics
    """
    lst: list[float]
    sum: float
    product: float

    def __init__(self) -> None:
        self.lst = []
        self.sum = 0.0
        self.product = 1.0

    def append(self, val: float) -> None:
        self.lst.append(val)
        self.sum += val
        self.product *= val

    def mean(self) -> float:
        return self.sum / len(self.lst)
```

Attributes: This is your wishlist for what you want your class to have in it.

Initialization method: creates your attributes and gives them a start value.

Methods: called using brackets. Acts like functions that can return values or update attributes.

Write a function `replace(a,low,high)` that:

1. Takes a Numpy array `a`
2. Replaces all values below `low` with `low`
3. Replaces all values above `high` with `high`
4. Return the mutated array

Create an array of temperatures. Using `np.where`, create a new array that stores values of 1 and 0. 1 will appear where temperatures are unsafe, for values warmer than 30 or cooler than -15. 0 will appear everywhere else. Save this new array in a file called "safe_temps.ssv".

Go to vevox.com

Sign in using the session ID: 127-293-257



Where

```
import numpy as np

a = np.array([-2, -1, 0, 1, 3])
b = np.where(a % 2 == 0, a * 10, a)
print(b)
```

What does this code print?

A. [-20 -1 0 1 3]

B. [-2 -10 0 10 30]

C. [-2 -1 0 1 3]

D. [-20 -10 0 10 30]



Where Concept

Which of the following statements about `np.where` is true?

- A. `np.where` always modifies the input array directly.
- B. `np.where(condition, x, y)` uses elements from `x` where the condition is True, and from `y` where it's False.
- C. `x` and `y` should only be integers
- D. `np.where(condition, x, y)` always returns the condition itself.



SSV

```
import numpy as np  
  
data = np.array([[1,2,3],  
                 [4,5,6]])
```

What is the correct way to save data?

A. np.savetxt(data, "output.ssv")

B. np.savetxt(data, "output.ssv")

C. np.savetxt("output.ssv", data)

D. np.savetxt("output.ssv", data)



Class

```
class StatsList:
    def __init__(self):
        self.lst = []
        self.sum = 0

    def append(self, v):
        self.lst.append(v)
        self.sum += v

s = StatsList()
s.append(4)
s.lst.append(10)
print(s.lst)
```

What is printed?

A. [4]

B. [14]

C. [4, 10]

D. error occurs



Class

```
class StatsList:
    def __init__(self):
        self.lst = []
        self.sum = 0

    def append(self, v):
        self.lst.append(v)
        self.sum += v

s = StatsList()
s.append(4)
s.lst.append(10)
print(s.sum)
```

What is printed?

A. 14

B. 4

C. 10

D. None



Class

```
class Dog:
    name: str

    def __init__(self, name: str) -> None:
        self.name = name

    def bark(self) -> str:
        return f"{self.name} says woof!"

buddy = Dog("Buddy")
print(buddy.bark())
```

What is printed?

A. "Buddy says woof!"

B. "Dog says woof!"

C. "self.name says woof!"

D. error occurs

Create a class called Student that has an attribute called grades: list[float]

- Initialize the list of grades to be empty
- Create an assignment method that will add a given assignment score to the list of grades
- Create an average method that will compute the current average and return it.

```
studentA = Student()
studentA.assignment(90.0)
studentA.assignment(88.5)
print("grades:", studentA.grades, "avg:", studentA.average())
```

grades: [90.0, 88.5] avg: 89.25