

Welcome to The Carpentries Etherpad!

This pad is synchronized as you type, so that everyone viewing this page sees the same text. This allows you to collaborate seamlessly on documents.

Use of this service is restricted to members of The Carpentries community; this is not for general purpose use (for that, try etherpad.wikimedia.org).

Users are expected to follow our code of conduct: https://docs.carpentries.org/topic_folders/policies/code-of-conduct.html

All content is publicly available under the Creative Commons Attribution License:
<https://creativecommons.org/licenses/by/4.0/>

Slides:

https://docs.google.com/presentation/d/1c48UUp8otd0phg4wJYO_ILM_4BWgg2keYzMY3t5S49w/edit?usp=sharing

Jupyter notebooks (.ipynb) for each of the lessons (with Challenge solutions !) are here:

<https://github.com/MonashDataFluency/python-workshop-base/tree/master/workshops/docs/modules/notebooks>

----- Mounting your Google Drive from inside Colaboratory ----

```
from google.colab import drive
drive.mount('/content/drive')
```

Now you can read/write files directly to the path '/content/drive' and they will appear in Google Drive.

```
string
string
```

```
str
str
str
str
```

```
tuple
tuple
tuple
tuple
```

```
C
c
c
```

c

Challenge: looping over a range

```
list(range(1,4))
list(range(1,4,1))
list(range(1,4,1))list(range(1,4))
```

```
for num in range(1, 4):
    print(num)
```

```
number = range(1,4)
for num in number:
    print(num)
```

```
for num in list(range(1,4)):
    print(num)
for num in list(range(1,4,1)):
    print (num)
```

Challenge - $5*5*5$ in a loop, using multiplication

```
a=1
for num in range(1,4):
    a*=5
print (a)
```

```
value=1
base=5
power=3
```

```
for times in range(power):
    value=value*base
```

```
print(value)
```

#First try (off for lunch).

#Issues:

```
# 1. Could not get the reverse order working
# 2. Loop to concatenate does not work properly
```

```
string1 = ['N', 'e', 'w', 't', 'o', 'n']
string_len=len(string1)
```

```

string_len1=len(string1)+1
for test in range(string_len, 0):
    name = name + string1[test]
print(name)

```

```

newt = {1:'N',2:'e',3:'w',4:'t',5:'o',6:'n'}
j=range(1,len(newt))
front='N'
back='n'
for n in j:
    front+=newt[n+1]
    p=len(j)-(n-1)
    back+=newt[p]
print(front)
print(back)

```

```

word = 'Newton'
b=0
C= len(word)
reverse=""

```

```

for num in range(C):
    b= len(word)-num-1
    reverse =reverse + word[b]
print(reverse)

```

<https://geoawesomeness.com/programming-arcgis-python-beginners-guide/>

1. names of columns, 2. no of rows and columns (shape of data file), 3.first 5 rows of data, 4. last 5 columns of data

```

site_names = pd.unique(surveys_df['site_id'])
24
none

```

```

site_names = pd.unique(surveys_df['site_id'])
len(site_names)

```

```

site_names = pd.unique(surveys_df['site_id'])
len(site_names)

```

```

site_names= pd.unique(surveys_df['site_id'])
print(site_names)
len(site_names)

```

```

site_names=pd.unique(df['site_id'])
spp=pd.unique(surveys_df['species_id'])
print(site_names)
print(spp)

```

```
print(len(site_names))
print(len(spp))
```

```
site_names= pd.unique(surveys_df['site_id'])
len(site_names)
```

```
group_data_sex.count() # - D
```

```
grouped_data=surveys_df.groupby('sex')
grouped_data.count() #answer = D
grouped_data['record_id'].describe()
```

The answer is D

You get the mean record_id, month, day, year hindfoot_length and weight for each site and gender

```
grouped_data3 = surveys_df.groupby(['site_id'])
grouped_data3['weight'].describe()
```

Challenge - first functions challenge

```
product_of_inputs = product(5,5)
# 25
product_of_inputs = product(2,5,3)
# TypeError: product() takes 2 positional arguments but 3 were given
```

```
def product(a, b):
    print('This function arguments are:', a, b, '(this is done inside the function!))')
    var = 3
    return a * b * var
product_of_inputs = product(2,5)
product_of_inputs
# 30
```

```
def product(a,b):
    print('The product of ',a,' and ',b,' is: ')
    return a * b
product(3423,442)
#The product of 3423 and 442 is:
1512966
```

---- Challenge - modifying functions

```
def year_to_csv_at_path(year, all_data, output_path, filename_prefix):

    # Select data for the year
    surveys_year = all_data[all_data.year == year]

    # Write the new DataFrame to a csv file
    filename = filename_prefix + str(year) + '.csv'
    filepath = output_path + '/' + filename

    if not os.path.exists(output_path):
        os.makedirs(output_path)

    surveys_year.to_csv(filepath)

def create_csvs_by_year_at_path(start_year,
                                end_year,
                                all_data,
                                output_path,
                                filename_prefix):
    # "end_year" is the last year of data we want to pull, so we loop to end_year+1
    for year in range(start_year, end_year+1):
        year_to_csv_at_path(year, all_data, output_path, filename_prefix)

def create_csvs_by_year_at_path2(start_year,
                                  end_year,
                                  all_data,
                                  output_path,
                                  filename_prefix):
    for year in range(start_year, end_year+1):
        # Select data for the year
        surveys_year = all_data[all_data.year == year]

        # Write the new DataFrame to a csv file
        filename = filename_prefix + str(year) + '.csv'
        filepath = output_path + '/' + filename

        if not os.path.exists(output_path):
            os.makedirs(output_path)

        surveys_year.to_csv(filepath)
    i

survs_df['site_id'] = survs_df['site_id'].astype('category')
ggplot(survs_df, aes(x='site_id', fill = 'species_id')) + \
geom_bar(stat="count")
```

