Programming for Social Scientists

Strings

Johan A. Dornschneider-Elkink

```
party = "Fianna Fáil"

abbreviation = 'FF'

intro = '''
Fianna Fáil was founded in 1926 by members of Sinn Féin, under
the leadership of Éamon de Valera. Since then, it has been one
of the leading political parties in the Republic of Ireland,
seeing its first serious vote collapse only 85 years later, in
the 2011 elections to Dáil Éireann.
'''
```

Creating strings

```python
def say_hello():
    """

    This function prints a friendly greeting to the screen.
    """


    print("Hello!")
```

```python
intro = '''
Fianna Fáil was founded in 1926 by members of Sinn Féin, under
the leadership of Éamon de Valera. Since then, it has been one
of the leading political parties in the Republic of Ireland,
seeing its first serious vote collapse only 85 years later, in
the 2011 elections to Dáil Éireann.
'''
```
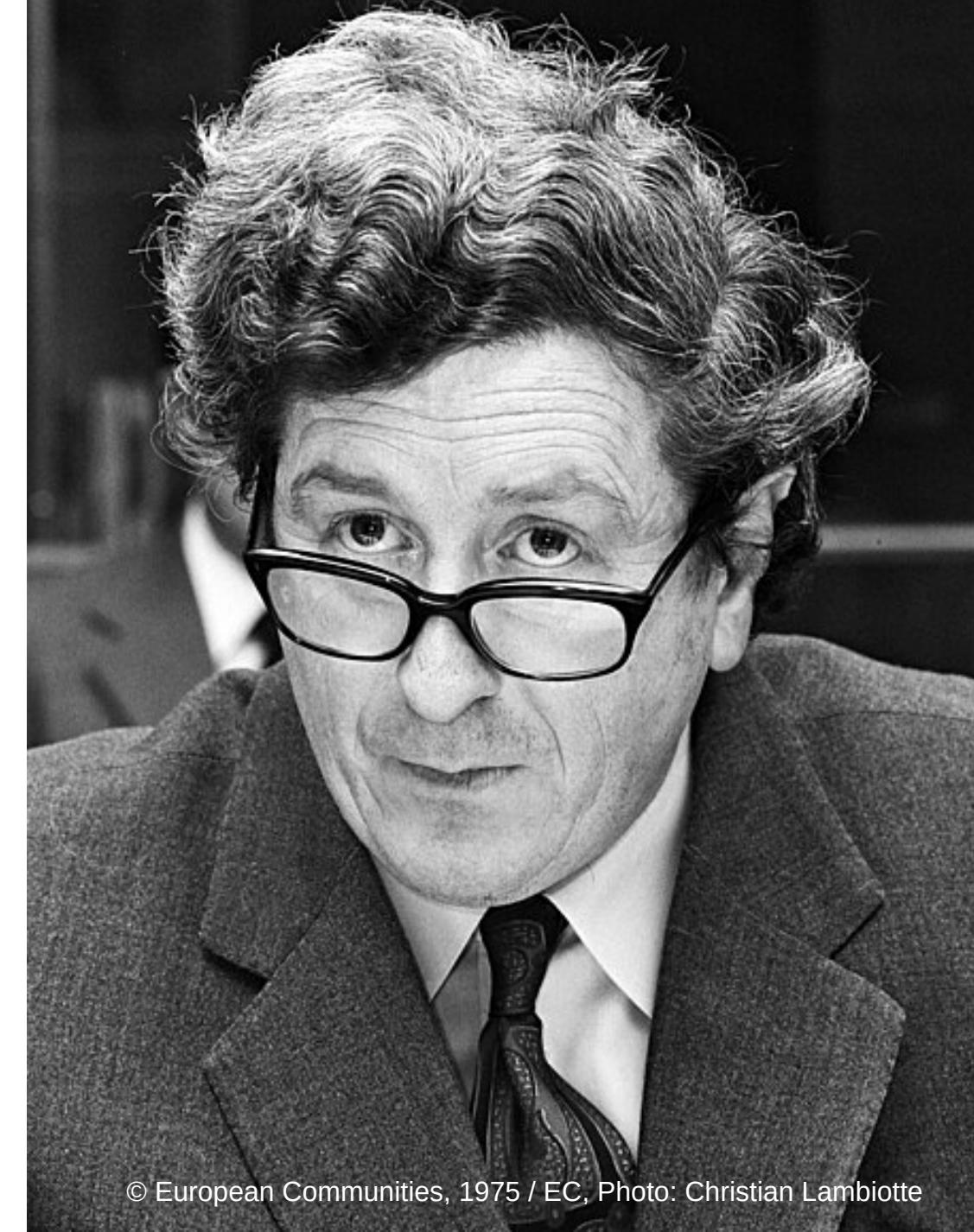
Creating strings

String indexing

```python
party = "Fine Gael"

first_character = party[0]
last_character = party[-1]

fine = party[:4]
gael = party[5:]
space = party[4:5]
```

Some typical string functions

```
party = "Sinn Féin"

title = '''
Mary Lou McDonald says Sinn Féin will re-run
referendums on family and care if they do not pass
'''

quote = '''
We would return to the Citizens' Assembly wording,
that's what should be happening now and if this is
not successful, that's where we will bring that.
'''

source = "Irish Independent, 20 Feb 2024"
```

Some typical string functions

```python
party = "Sinn Féin"

title = '''
Mary Lou McDonald says Sinn Féin will re-run
referendums on family and care if they do not pass
'''

quote = '''
We would return to the Citizens' Assembly wording,
that's what should be happening now and if this is
not successful, that's where we will bring that.
'''

source = "Irish Independent, 20 Feb 2024"
```

```python
number_of_characters = len(quote)

quote_lowercase = quote.lower()
quote_uppercase = quote.upper()

count_references_to_we = quote.lower().count("we")

replace_we = quote.lower().replace("we", "they")
```

Some typical string functions

```python
party = "Sinn Féin"

title = '''
Mary Lou McDonald says Sinn Féin will re-run
referendums on family and care if they do not pass
'''
```

```
                                                    ssembly wording,
                                                    w and if this is
                                                    ll bring that.

                                                    o 2024"
```

```python
def remove_punctuation(s):
    """
    Returns string s with all punctuation removed.
    """


list_of_words = remove_punctuation(quote)
```

Some typical string functions

```python
party = "Sinn Féin"

title = '''
Mary Lou McDonald says Sinn Féin will re-run
referendums on family and care if they do not pass
'''
```

```python
import string


def remove_punctuation(s):
    """
    Returns string s with all punctuation removed.
    """

    translator = str.maketrans("", "", string.punctuation)

    return s.translate(translator)


list_of_words = remove_punctuation(quote)
```

This is just an example – don't worry about the technical details!

...ssembly wording,
...w and if this is
...ll bring that.

...o 2024"

Some typical string functions

```python
party = "Sinn Féin"

title = '''
Mary Lou McDonald says Sinn Féin will re-run
referendums on family and care if they do not pass
'''
```

This is just an example - don't worry about the technical details!

...ssembly wording,
...w and if this is
...ll bring that.

...2024"

```python
import string


def remove_punctuation(s):
    """

    Returns string s with all punctuation removed.
    """

    translator = str.maketrans("", "", string.punctuation)


    return s.translate(translator)


list_of_words = remove_punctuation(quote).lower().replace("\n", "").split(" ")
list_of_words.sort()


print(", ".join(list_of_words))
```

Some typical string functions

```python
party = "Sinn Féin"

title = '''
Mary Lou McDonald says Sinn Féin will re-run
referendums on family and care if they do not pass
'''
```

```python
import string

def remove_punctuation(s):
    """
    Returns string s with all punctuation removed.
    """

    translator = str.maketrans("", "", string.punctuation)

    return s.translate(translator)


list_of_words = remove_punctuation(quote).lower().replace("\n", "").split(" ")
list_of_words.sort()

print(", ".join(list_of_words))
```

```
and, assembly, be, bring, citizens, happening, if, is, not,
 now, return, should, successful, that, thats, thats, the,
this, to, we, we, what, where, will, wording, would
```

Adding numbers to strings

```python
party = "Green Party"
founded = 1981
age = 2023 - 1981
percent_1st_pref = 7.1
```

The Green Party, founded in 1981, is now 42 years old. In the most recent elections, it won 7.1 % of the first preference votes.

Adding numbers to strings (method 1)

```python
print("The %s, founded in %d, is now %d years
old. In the most recent elections, it won %.1f
%% of the first preference votes." % (party,
founded, age, percent_1st_pref))
```

```python
party = "Green Party"
founded = 1981
age = 2023 - 1981
percent_1st_pref = 7.1
```

The Green Party, founded in 1981, is now 42 years old. In the most recent elections, it won 7.1 % of the first preference votes.

Adding numbers to strings (method 1)

%d   Integer

%f   Floating point

%s   String

%x   Hexadecimal

%%   Percentage sign

%3d   Integer, using 3 digits

%03d   Integer, using 3 digits, padding with zeros

%5f   Floating point, using 5 digits

%5.2f   Floating point, using 5 digits, and rounding
          to 2 digits after decimal point

%10s   String, 10 characters long

```python
print("The %s, founded in %d, is now %d years
old. In the most recent elections, it won %.1f
%% of the first preference votes." % (party,
founded, age, percent_1st_pref))
```

```python
party = "Green Party"
founded = 1981
age = 2023 - 1981
percent_1st_pref = 7.1
```

The Green Party, founded in 1981, is now 42 years old. In the most recent elections, it won 7.1 % of the first preference votes.

Adding numbers to strings (method 2)

```python
print(f"The {party}, founded in {founded}, is
now {age} years old. In the most recent
elections, it won {percent_1st_pref} % of the
first preference votes.")
```

Note the little f in front ...

```python
party = "Green Party"
founded = 1981
age = 2023 - 1981
percent_1st_pref = 7.1
```

The Green Party, founded in 1981, is now 42 years old. In the most recent elections, it won 7.1 % of the first preference votes.

Adding numbers to strings (method 3)

```python
print("The {}, founded in {}, is now {} years
old. In the most recent elections, it won {} %
of the first preference votes.".format(party,
founded, age, percent_1st_pref))
```

```python
party = "Green Party"
founded = 1981
age = 2023 - 1981
percent_1st_pref = 7.1
```

Detailed formatting is possible:

```python
print("The {:s}, founded in {:d}, is now {:d}
years old. In the most recent elections, it won
{:.1f} % of the first preference votes.".format
(party, founded, age, percent_1st_pref))
```

The Green Party, founded in 1981, is now 42 years old. In the most recent elections, it won 7.1 % of the first preference votes.

Input from the user

```python
name = input("Please enter your name: ")

age = int(input("Please enter your age: "))
```

Input from the user

```python
name = input("Please enter your name: ")

age = int(input("Please enter your age: "))
```

"Gambler" by dariuszka

```python
import random

guess = int(input("Enter your guess: "))
selection = ""

while selection != "q":

    print('''
    Main menu

    (E)nter new guess
    (T)hrow dice
    (Q)uit program

    ''')

    selection = input("Enter your menu choice: ")

    if selection == "t":
        print(random.randint(1,6) == guess)
    elif selection == "e":
        guess = int(input("Enter your guess: "))
```