**Getting Started**

Let's get started and try out spaCy! In this exercise, you'll be able to try out some of the 45+ [available languages](https://spacy.io/usage/models#languages).

*This course introduces a lot of new concepts, so if you ever need a quick refresher, download the* [*spaCy Cheat Sheet*](http://datacamp-community-prod.s3.amazonaws.com/29aa28bf-570a-4965-8f54-d6a541ae4e06) *and keep it handy!*

**Instructions 1/3**

35 XP

* [1](javascript:void(0))
  + Import the English class from spacy.lang.en and create the nlp object.
  + Create a doc and print its text.

# Import the English language class

from spacy.lang.en import English

# Create the nlp object

nlp = English()

# Process a text

doc = nlp("This is a sentence.")

# Print the document text

print(doc.text)

* Import the German class from spacy.lang.de and create the nlp object.
* Create a doc and print its text.

# Import the German language class

from spacy.lang.de import German

# Create the nlp object

nlp = German()

# Process a text (this is German for: "Kind regards!")

doc = nlp("Liebe Grüße!")

# Print the document text

print(doc.text)

* Import the Spanish class from spacy.lang.es and create the nlp object.
* Create a doc and print its text.

# Import the Spanish language class

from spacy.lang.es import Spanish

# Create the nlp object

nlp = Spanish()

# Process a text (this is Spanish for: "How are you?")

doc = nlp("¿Cómo estás?")

# Print the document text

print(doc.text)

<script.py> output:

This is a sentence.

<script.py> output:

Liebe Grüße!

<script.py> output:

¿Cómo estás?

**Documents, spans and tokens**

When you call nlp on a string, spaCy first tokenizes the text and creates a document object. In this exercise, you'll learn more about the Doc, as well as its views Token and Span.

**Instructions 1/2**

50 XP

* [1](javascript:void(0))
* [2](javascript:void(0))
* Import the English language class and create the nlp object.
* Process the text and instantiate a Doc object in the variable doc.
* Select the first token of the Doc and print its text.

# Import the English language class and create the nlp object

from spacy.lang.en import English

nlp = English()

# Process the text

doc = nlp("I like tree kangaroos and narwhals.")

# Select the first token

first\_token = doc[0]

# Print the first token's text

print(first\_token.text)

<script.py> output:

I

**Instructions 2/2**

50 XP

* [2](javascript:void(0))
* Create a slice of the Doc for the tokens "tree kangaroos" and "tree kangaroos and narwhals".

# Import the English language class and create the nlp object

from spacy.lang.en import English

nlp = English()

# Process the text

doc = nlp("I like tree kangaroos and narwhals.")

# A slice of the Doc for "tree kangaroos"

tree\_kangaroos = doc[2:4]

print(tree\_kangaroos.text)

# A slice of the Doc for "tree kangaroos and narwhals" (without the ".")

tree\_kangaroos\_and\_narwhals = doc[2:-1]

print(tree\_kangaroos\_and\_narwhals.text)

<script.py> output:

tree kangaroos

tree kangaroos and narwhals

**Lexical attributes**

In this example, you'll use spaCy's Doc and Token objects, and lexical attributes to find percentages in a text. You'll be looking for two subsequent tokens: a number and a percent sign. The English nlp object has already been created.

**Instructions**

100 XP

* Use the like\_num token attribute to check whether a token in the doc resembles a number.
* Get the token *following* the current token in the document. The index of the next token in the doc is token.i + 1.
* Check whether the next token's text attribute is a percent sign "%".

# Process the text

doc = nlp("In 1990, more than 60% of people in East Asia were in extreme poverty. Now less than 4% are.")

# Iterate over the tokens in the doc

for token in doc:

# Check if the token resembles a number

if token.like\_num:

# Get the next token in the document

next\_token = doc[token.i + 1]

# Check if the next token's text equals '%'

if next\_token.text == '%':

print('Percentage found:', token.text)

<script.py> output:

Percentage found: 60

Percentage found: 4

Well done! As you can see, you can do a lot of very powerful analyses using the tokens and their attributes.