In [1]:
```
import spacy
nlp = spacy.load('en_core_web_sm')
```

In [2]:
```
def show_ents(doc):
    if doc.ents:
        for ent in doc.ents:
            print(ent.text + ' - ' + ent.label_ + ' - ' + str(spacy.expl
    else:
        print('No entities found!')
```

In [3]:
```
doc = nlp(u'Hi! how are you?')
```

In [4]:
```
show_ents(doc)
```

```
No entities found!
```

In [5]:
```
doc = nlp(u"May I go to Washington, DC next May to see the Washington Mo
```

In [6]:
```
show_ents(doc)
```

```
Washington, DC - GPE - Countries, cities, states
next May - DATE - Absolute or relative dates or periods
the Washington Monument - ORG - Companies, agencies, institutions, etc
.
```

In [7]:
```
doc = nlp(u"Can I please have 500 dollars of Microsoft stock")
```

In [8]:
```
show_ents(doc)
```

```
500 dollars - MONEY - Monetary values, including unit
Microsoft - ORG - Companies, agencies, institutions, etc.
```

Adding a Named Entity to a Span

In [9]:
```
doc = nlp("Tesla to build a new U.K. factory for $6 million")
```

In [10]:
```
show_ents(doc)
```

```
U.K. - GPE - Countries, cities, states
$6 million - MONEY - Monetary values, including unit
```

In [11]:
```
from spacy.tokens import Span
```

In [12]:
```
ORG = doc.vocab.strings[u'ORG']
```

In [13]:
```
ORG
```

Out[13]:
```
381
```

In [14]:
```
# Create a span for the new entity
new_ent = Span(doc, 0, 1, label=ORG)
```

In [15]:
```
doc.ents = list(doc.ents) + [new_ent]
```

In [16]:
```
show_ents(doc)
```

```
Tesla - ORG - Companies, agencies, institutions, etc.
U.K. - GPE - Countries, cities, states
$6 million - MONEY - Monetary values, including unit
```

Adding Named Entities to All Matching Spans

In [17]:
```
# We want to add vacuum cleaner and vacuum-cleaner as PROD(product) NERs
```

In [18]:
```
doc = nlp(u"Our company created a brand new vacuum cleaner."
          u"This new vacuum-cleaner is the best in the show.")
```

In [19]:
```
# Check the entities in the document
show_ents(doc)
```

```
No entities found!
```

In [20]:
```
from spacy.matcher import PhraseMatcher
```

In [21]:
```
matcher = PhraseMatcher(nlp.vocab)
print(matcher)
```

```
<spacy.matcher.PhraseMatcher object at 0x11c744c90>
```

In [22]:
```
phrase_list = ['vacuum cleaner', 'vacuum-cleaner']
```

In [23]:
```
phrase_patterns = [nlp(text) for text in phrase_list]
print(phrase_patterns)
```

```
[vacuum cleaner, vacuum-cleaner]
```

In [24]:
```
matcher.add('newproduct', None, *phrase_patterns)
```

In [25]:
```
found_matches = matcher(doc)
```

In [26]:
```
print(found_matches)
```

```
[(2689272359382549672, 6, 8), (2689272359382549672, 11, 14)]
```

In [27]:
```
from spacy.tokens import Span
```

In [28]:
```
PROD = doc.vocab.strings[u"PRODUCT"]
```

In [31]:
```
new_ents = [Span(doc, match[1],match[2], label=PROD) for match in found_
```

In [32]:
```
print(new_ents)
```

```
[vacuum cleaner, vacuum-cleaner]
```

In [33]:
```
doc.ents = list(doc.ents) + new_ents
```

In [34]:
```
print(doc.ents)
```

```
(vacuum cleaner, vacuum-cleaner)
```

In [35]:
```
show_ents(doc)
```

```
vacuum cleaner - PRODUCT - Objects, vehicles, foods, etc. (not service
s)
vacuum-cleaner - PRODUCT - Objects, vehicles, foods, etc. (not service
s)
```

In [36]:
```
# Example 2 -- Money Entity
```

In [37]:
```
doc = nlp(u"Originally I paid $29.95 for this toy car, but now it is mar
```

In [38]:
```
# to look at all entities
[ent for ent in doc.ents]
```

Out[38]:
```
[29.95, 10 dollars]
```

In [39]:
```
show_ents(doc)
```

```
29.95 - MONEY - Monetary values, including unit
10 dollars - MONEY - Monetary values, including unit
```

In [40]:
```
[ent for ent in doc.ents if ent.label_ == 'MONEY']
```

Out[40]:
```
[29.95, 10 dollars]
```

In [41]:
```
# To find how many entities
len([ent for ent in doc.ents if ent.label_ == 'MONEY'])
```

Out[41]:
```
2
```

In [ ]: