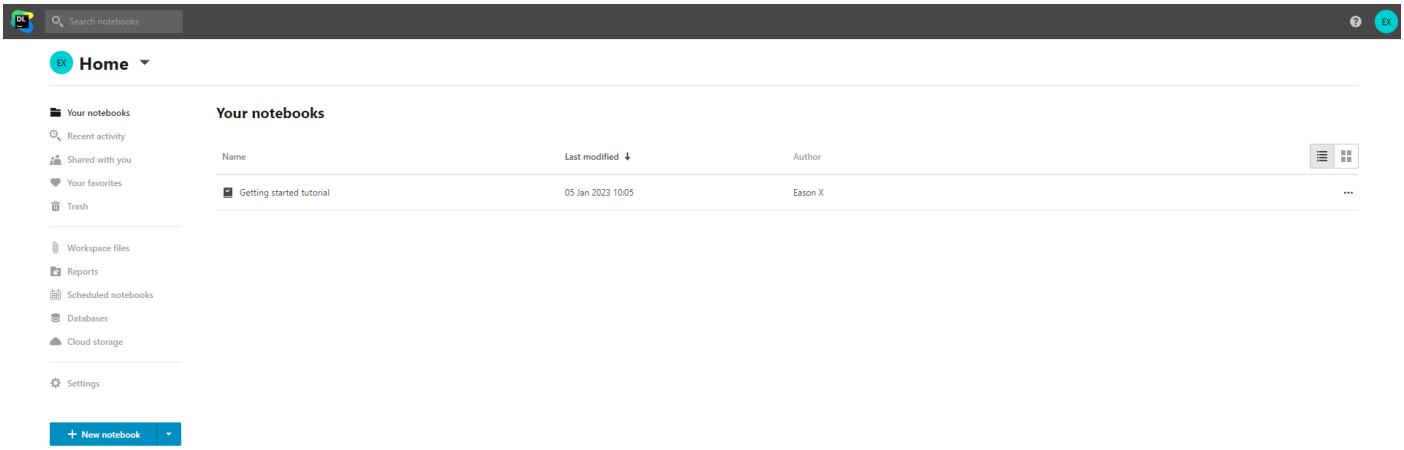
# Part Two. Virtual Assistant

## How to Open JetBrains Datalore for Virtual Assistant

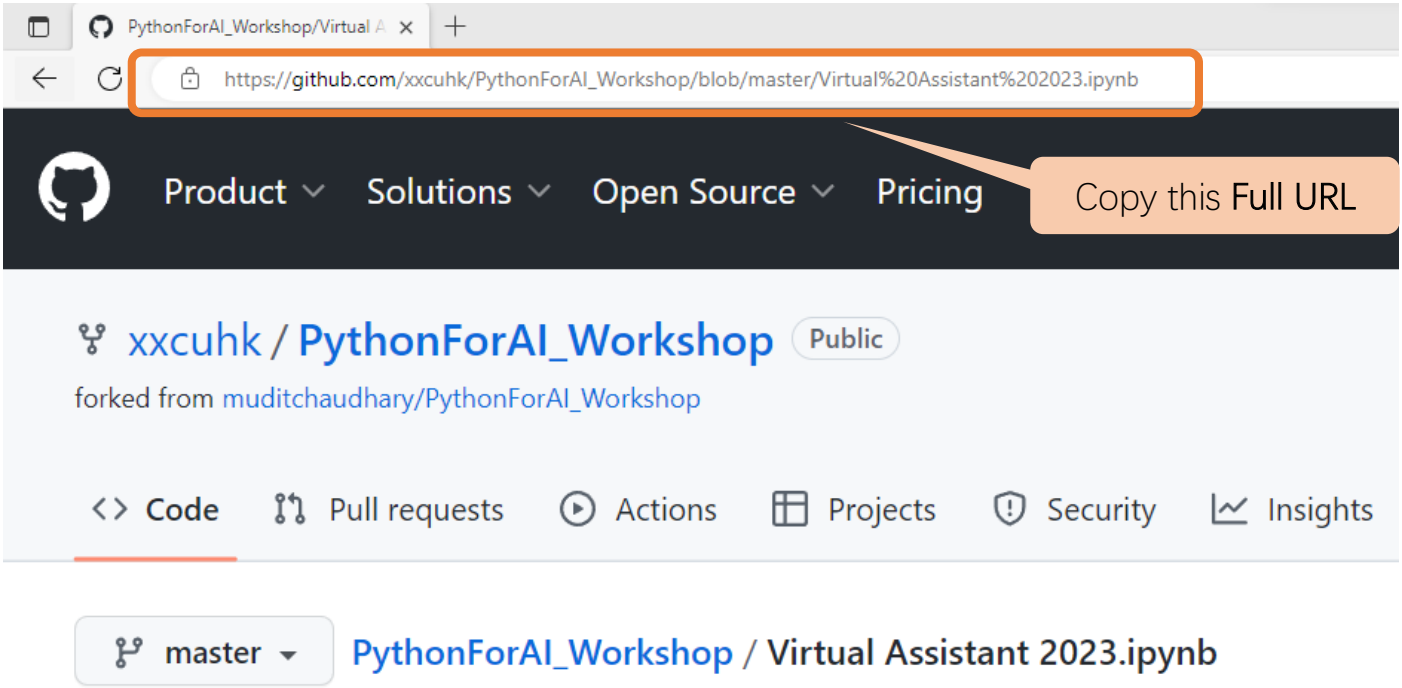
### Step 1. Registration

1. Access and sign in to JetBrains **Datalore** Platform using the following link: datalore.jetbrains.com
2. Once logged in, the interface will be displayed as shown below:

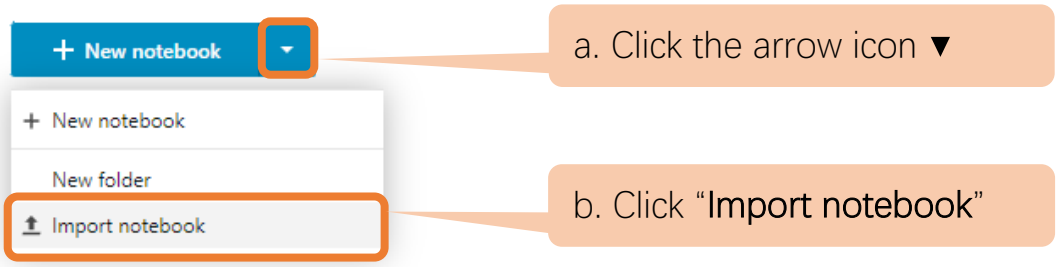



### Step Two. Import Notebook

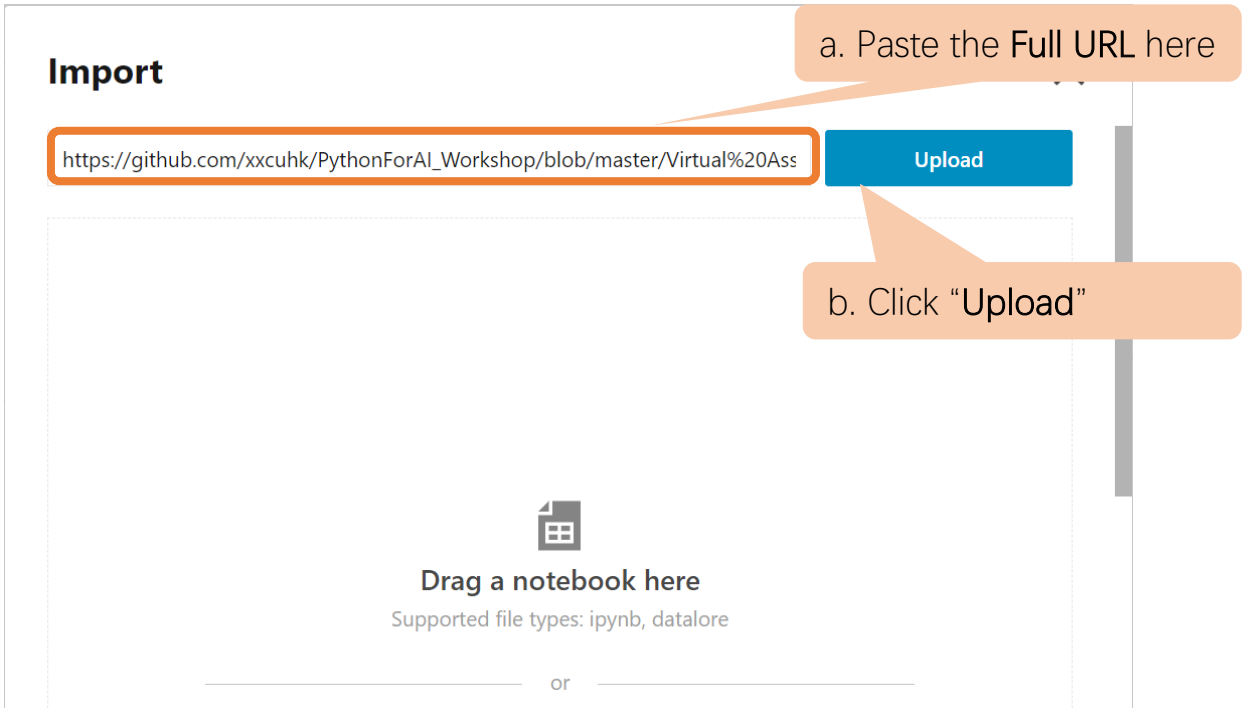
1. Open a **new tab** in your browser and key in cutt.ly/i3kCSC8 in the URL bar. Copy the **full URL**.



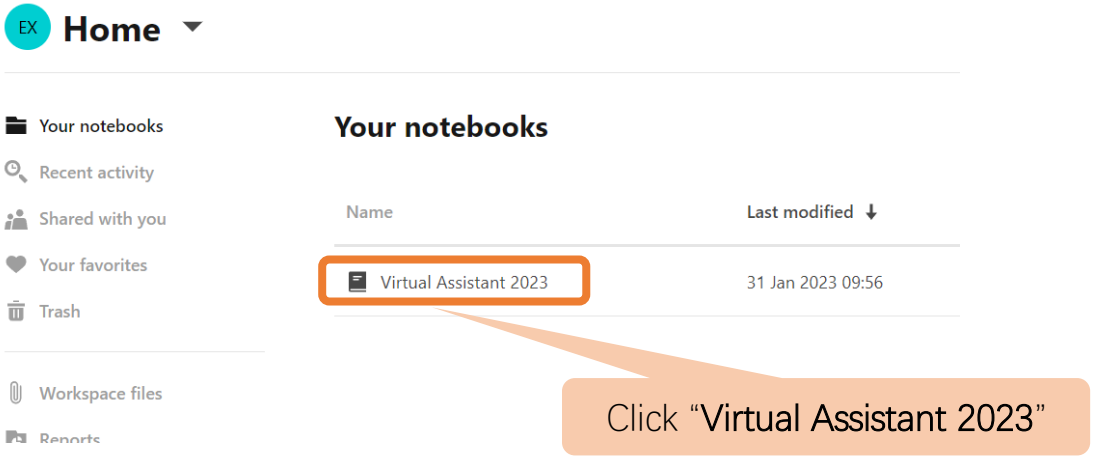
2. Press the arrow ▼ beside **'New notebook'** and click **'Import notebook'**.




3. Paste the **full URL** and click "**Upload**".




4. Once uploaded, click "**Virtual Assistant 2023**".




5. Please refer to the following slides for creating your virtual assistant.

# 2.1.1 Install and Import libraries

## 2.1.1 Install and Import libraries

Step 1. Click **run**.

Run the cell below to install and import the required libraries and functions.

**Note:** *We have pre-written some code to simplify the weath... and movie rating APIs in the file* `utils.py`. *You can view the file later to understand the inner working in more detail.*

```
# Run this cell
!pip install --upgrade pip
!pip install pyjokes
!pip install snips-nlu
!pip install pyowm
!pip install imdbpy
!python -m snips_nlu download en
!git clone https://github.com/xxcuhk/workshop_utils
from workshop_utils.utils import *
import pyjokes
import json
from snips_nlu import SnipsNLUEngine
from snips_nlu.default_configs import CONFIG_EN
```

# 2.1.1 Install and Import libraries

[1]

```
# Run this cell
!pip install --upgrade pip
!pip install pyjokes
!pip install snips-nlu
!pip install pyowm
!pip install imdbpy
!python -m snips_nlu download en
!git clone https://github.com/xxcuhk/workshop_utils
from workshop_utils.utils import *
import pyjokes
import json
from snips_nlu import SnipsNLUEngine
from snips_nlu.default_configs import CONFIG_EN
```

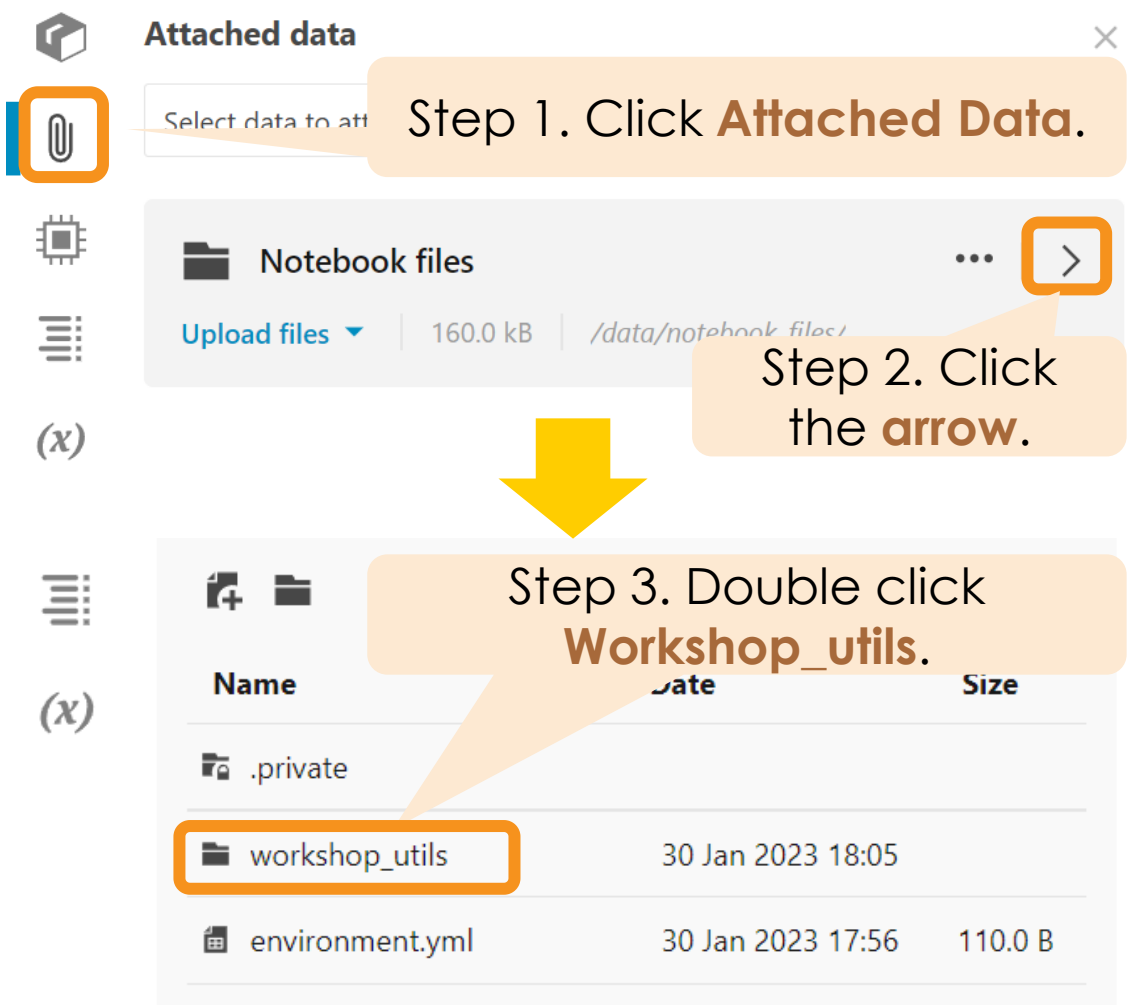
+ Show all

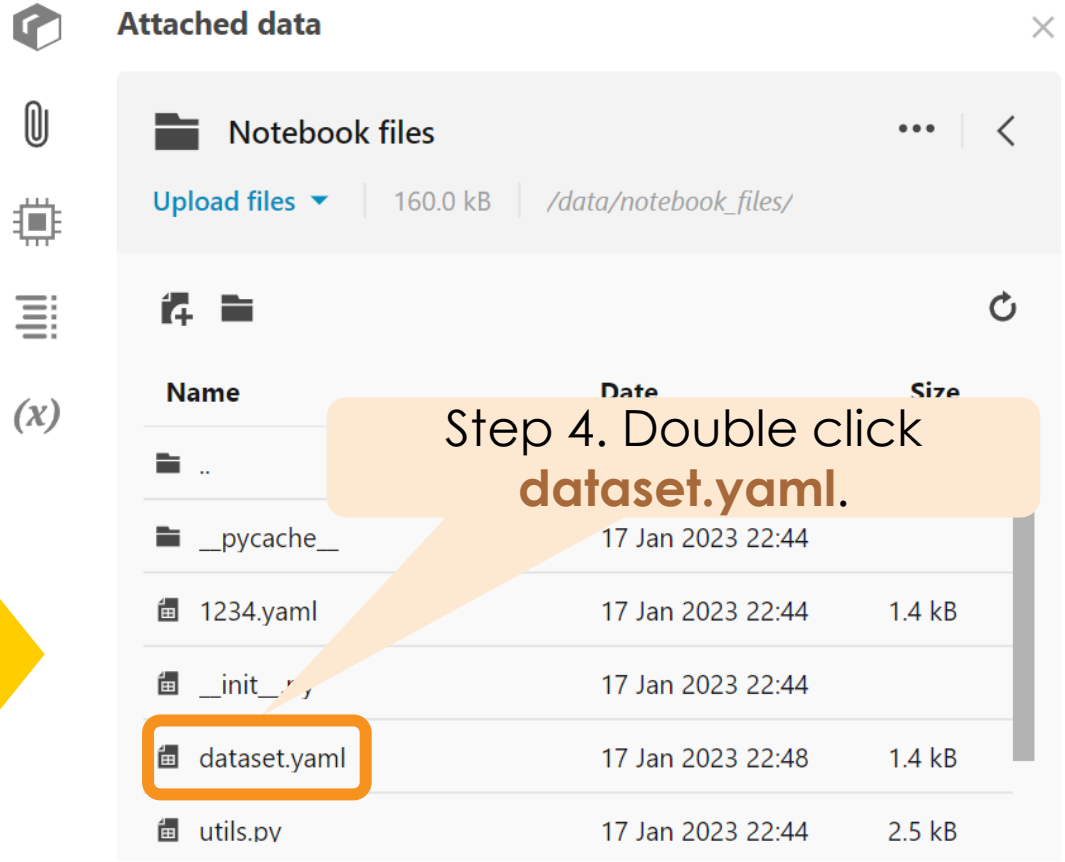
```
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.5/1.5 MB 20.4 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... - done
Building wheels for collected packages: snips_nlu_en
  Building wheel for snips_nlu_en (setup.py) ... - \ | done
  Created wheel for snips_nlu_en: filename=snips_nlu_en-0.2.3-py3-none-any.whl size=1297478 sha256=26aac5ae74f4bbe4efcf5ffbe69c7d2bc
  Stored in directory: /tmp/pip-ephem-wheel-cache-5vz3es49/wheels/77/e5/27/a2c7ae7b04c836360914a1ac909339da898cb66444e709f650
Successfully built snips_nlu_en
Installing collected packages: snips_nlu_en
Successfully installed snips_nlu_en-0.2.3

Linking successful
/opt/python/envs/default/lib/python3.8/site-packages/snips_nlu_en/snips_nlu_en-0.2.3 --> /opt/python/envs/default/lib/python3.8/site

Cloning into 'workshop_utils'...
remote: Enumerating objects: 50, done.
remote: Counting objects: 100% (50/50), done.
remote: Compressing objects: 100% (38/38), done.
remote: Total 50 (delta 18), reused 42 (delta 11), pack-reused 0
Unpacking objects: 100% (50/50), 9.08 KiB | 16.00 KiB/s, done.
```

This is the output.

Now, you are ready to prepare the training dataset

# 2.1.2 Create Training Dataset



# 2.1.2 Create Training Dataset

Intent 1
**Finished**

```
# tell_joke intent
---
type: intent
name: tell_joke

utterances:
  - Hi, tell me a joke.
  - I'm bored, entertain me with a joke
```

A special case without slots

Intent 2
**Finished**

```
# get_weather intent
---
type: intent
name: get_weather
slots:
  - name: city
    entity: city

utterances:
  - How is the weather in [city](London)?
  - What is the weather in [city](New York)?
  - Can you tell me how is the weather like in [city](Hong Kong)?
  - I wonder how hot is it in [city](Paris)?
```

Use this as an example of intent definition.

# 2.1.2 Create Training Dataset

Intent 3
Not finished

```
# get_rating intent
---
type: intent
name: get_rating
slots:
  - name:
    entity:

utterances:
  -
  -
```

Please complete it following the syntax of **intent 2**.

Intent 3
**Finished**

```
# get_rating intent
---
type: intent
name: get_rating
slots:
  - name: movie_name
    entity: movie_name

utterances:
  - How good is the movie [movie_name](Batman)?
  - I want to know the movie rating for [movie_name](Star Wars).
```

# 2.1.2 Create Training Dataset

Intent 4
Not finished

```
# get_director intent
---
type: intent
name: get_director
slots:
  - name:
    entity:

utterances:
  -
  -
```

Please complete it following the syntax of **intent 2**.

Intent 4
**Finished**

```
# get_director intent
---
type: intent
name: get_director
slots:
  - name: movie_name
    entity: movie_name

utterances:
  - Who directed [movie_name](Tenet)?
  - I want to know the director of the movie [movie_name](Ip Man).
```

You may input more utterances for training a better model.

# 2.1.2 Create Training Dataset

Entity 1
**Finished**

```
# city entity
---
type: entity
name: city
values:
  - Hong Kong
  - New York
  - Paris
  - London
  - Tokyo
  - Shanghai
```

Please use **Entity 1** as an example of intent definition.

Please keep the same indentation.

# 2.1.2 Create Training Dataset

Entity 2
Not finished

```
# movie_name entity
---
type: entity
name: movie_name
values:
  -
  -
  -
  -
```

Please complete it following the syntax of **entity 1**.

Entity 2
**Finished**

```
# movie_name entity
---
type: entity
name: movie_name
values:
  - Star Wars
  - Spider-man
  - Titanic
  - The Others
```

You may input more values for training a better model.

# 2.1.3 Convert the Dataset to json Format

Run the next cell to convert the dataset to json format to train the NLU Engine.

Step 1. click **Run**.

```
1.3s
# Run this cell

!snips-nlu generate-dataset en workshop_utils/dataset.yaml > dataset.json
```

There is no output at this stage.

# 2.1.4 Open the Dataset

To open the dataset, we will follow the following steps:

1. Use `open` function to load the file into Python in a variable called `dataset_file`.

   To do this you need to run: `dataset_file = open("dataset.json", "r")`

2. Use `load` function from `json` as `json.load(dataset_file)` into a variable called `training_dataset`.

   To do this you need to run: `training_dataset = json.load(dataset_file)`

```
0.1s
# Write the code below
dataset_file = open("dataset.json", "r")
training_dataset = json.load(dataset_file)
```

Step 1.
**Write the code**.
Step 2.
Click **run**.

There is no output at this stage.

# 2.2.1 Initialize the Snips-NLU Engine with English Configuration

We will start our Snips-NLU engine using the `SnipsNLUEngine()` function. `config=CONFIG_EN` will load the English language configuration in our NLU engine.

The Snips-NLU engine will be saved in a variable called `NLUengine`.

To do this you need to run: `NLUengine = SnipsNLUEngine(config=CONFIG_EN)`

```
▷ 0.1s
# Write the code below
NLUengine = SnipsNLUEngine(config=CONFIG_EN)
```

Step 1.
**Write the code**.
Step 2.
Click **run**.

There is no output at this stage.

**Note:**
Python is case sensitive language.

# 2.2.2 Train the NLU Engine

We will now train the NLU engine using our training dataset. `fit()` function is called to train the model.

To do this you need to run: `NLUengine.fit(training_dataset)`

```
4.1s
# Write the code below

NLUengine.fit(training_dataset)
```

```
<snips_nlu.nlu_engine.nlu_engine.SnipsNLUEngine at 0x7f4a8a7d8490>
```

Step 1.
**Write the code**.
Step 2.
Click **run**.

This is the correct output.

# 2.2.3 Use the NLU Engine to Parse the Intention

Let's try to use our engine on the utterance `"How's the weather in Hong Kong"`

Use the function `prediction = NLUengine.parse(your utterance)`

```
0.0s
# Write the code below

prediction = NLUengine.parse("How is the weather in Hong Kong?")
```

There is no output at this stage.

Step 1.
**Write the code** and **key in your utterance**.
Step 2.
Click **run**.

# 2.2.4 Print the Prediction

To print the prediction in a more readable format we will use `json.dumps()` function as:

```
print(json.dumps(prediction, indent=2))
```

0.1s

```
# Write the code below
print(json.dumps(prediction, indent=2))
```

+ Show all


```
    "intentName": "get_weather",
    "probability": 1.0
  },
  "slots": [
    {
      "range": {
        "start": 22,
        "end": 31
      },
      "rawValue": "Hong Kong",
      "value": {
        "kind": "Custom",
        "value": "Hong Kong"
      },
      "entity": "city",
      "slotName": "city"
    }
  ]
}
```

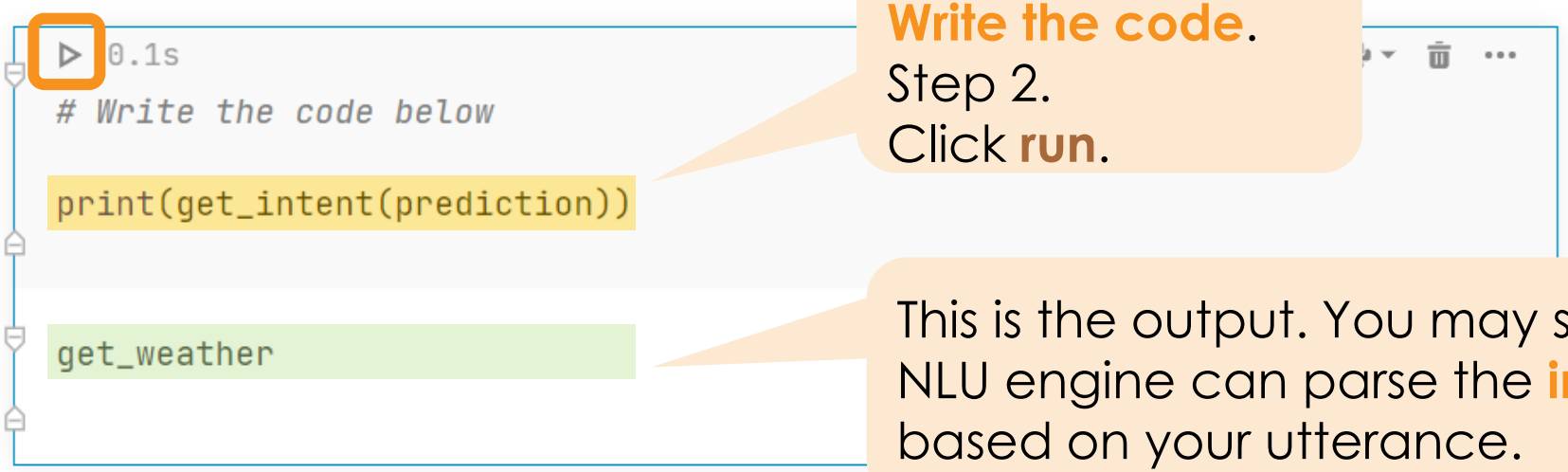

Step 1.
**Write the code**.
Step 2.
Click **run**.

This is the output.

# 2.2.5 Get the Intent

To get the intent we access the intent name element from the resulted `prediction` dictionary.

We have made a function for you to get the intent easily. You can use `get_intent(prediction)` to get the intent.

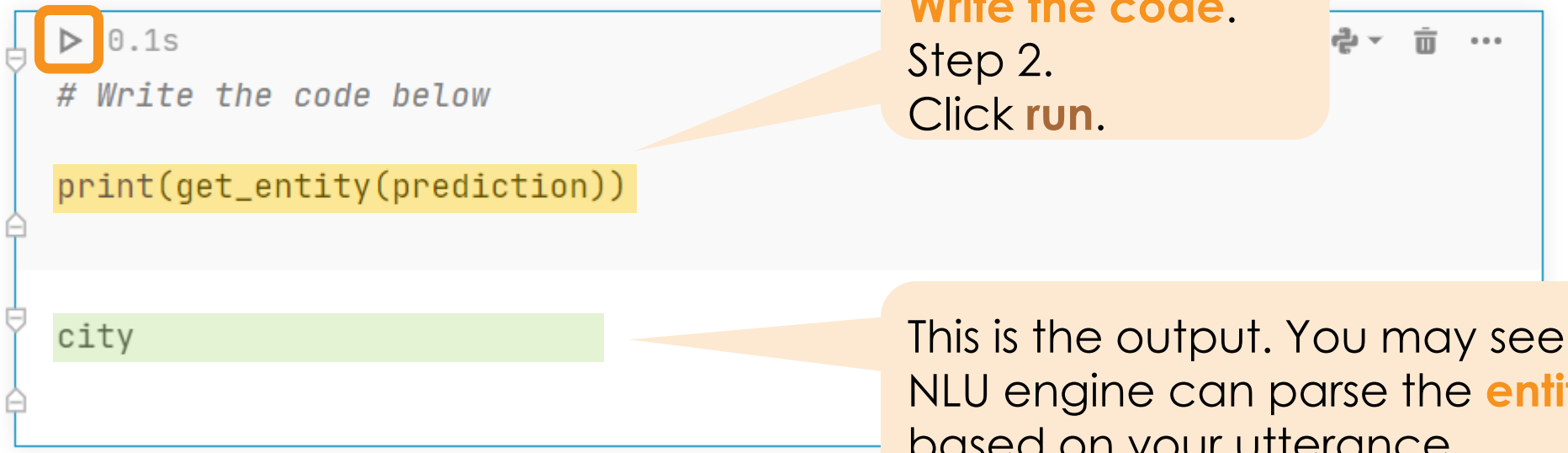



Step 1.
**Write the code**.
Step 2.
Click **run**.

This is the output. You may see if the NLU engine can parse the **intention** based on your utterance.

# 2.2.6 Get the Entity

You can use our function `get_entity(prediction)` to get the slot's entity.

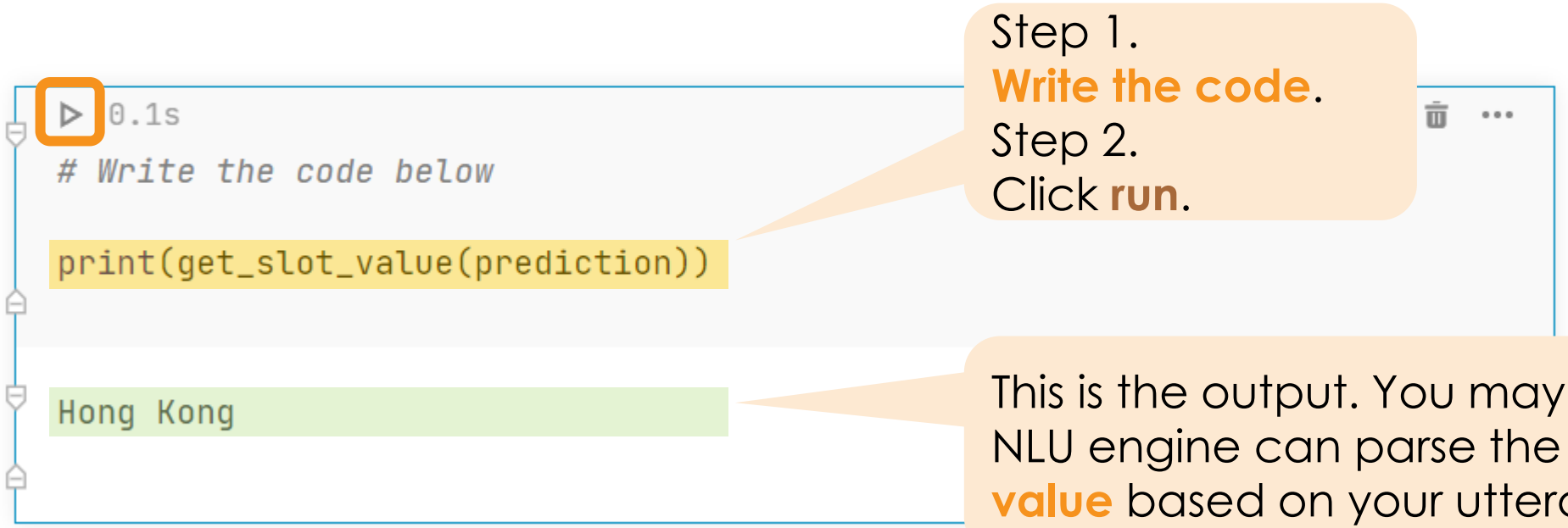



Step 1.
**Write the code**.
Step 2.
Click **run**.

This is the output. You may see if the NLU engine can parse the **entity** based on your utterance.

# 2.2.7 Get the Slots's Value

You can use our function `get_slot_value(prediction)` to get the slot's value.




Step 1.
**Write the code**.
Step 2.
Click **run**.

This is the output. You may see if the NLU engine can parse the **slots's value** based on your utterance.

# 2.3.1 Create a Function

1. Define a function called `assistant`.
2. The function has a parameter called `utterance`.

```
def assistant("""Enter the required Parameter here"""):
    prediction =
    intent =
    entity =
```

1. Use the NLU Engine to parse the intention.
2. You can use `get_intent(prediction)` to get the intent.
3. Use our function `get_entity(prediction)` to get the slot's entity.

```
        iction)
        else:
            print("Sorry, can you try again?")

    elif (intent == "get_rating"):

        if (entity == "movie_name"):
            movie_name = get_slot_value(prediction)
            get_movie_rating(movie_name)
        else:
            print("Sorry, can you try again?")

    elif (intent == "get_director"):



    else:
        print("Unknown intent")
```

Complete it following the syntax of the condition `get_rating` intent.

```
def assistant(utterance):
    prediction = NLUengine.parse(utterance)
    intent = get_intent(prediction)
    entity = get_entity(prediction)



    if (intent == "tell_joke"):
        print(pyjokes.get_joke())

    elif (intent == "get_weather"):

        if (entity == "city"):
            city_name = get_slot_value(prediction)
            get_city_weather(city_name)
        else:
            print("Sorry, can you try again?")

    elif (intent == "get_rating"):

        if (entity == "movie_name"):
            movie_name = get_slot_value(prediction)
            get_movie_rating(movie_name)
        else:
            print("Sorry, can you try again?")

    elif (intent == "get_director"):

        if (entity == "movie_name"):
            movie_name = get_slot_value(prediction)
            get_movie_directors(movie_name)
        else:
            print("Sorry, can you try again?")

    else:
        print("Unknown intent")
```

# 2.3.2 Create a Conversation Loop

```
print("Welcome to the virtual assistant. How can I help you?")
while True:
    print("------------------------------------")
    user_input=str(input("Enter your input: "))

    # This if statement should break the loop if the user_input is "Bye"
    if (user_input == "Bye" or user_input == "bye"):
        print("Have a good day!")
        break

    else:
        print("Assistant: ")
        """Enter the code here to call assistant function using user_input here"""
```

1. Call the `assistant` function that we have defined in 2.3.1.
2. The parameter should be `user's input`.

```
print("Welcome to the virtual assistant. How can I help you?")
while True:
    print("------------------------------------")
    user_input=str(input("Enter your input: "))

    # This if statement should break the loop if the user_input is "Bye"
    if (user_input == "Bye" or user_input == "bye"):
        print("Have a good day!")
        break

    else:
        print("Assistant: ")
        assistant(user_input)
```

# Chat with Virtual Assistant

Let's type some questions to chat with your Virtual Assistant!

You may input **bye** or **Bye** to quit the chatting.

```
Welcome to the virtual assistant. How can I help you?
-------------------------------------
Enter your input:  Tell me a joke
Assistant:
How many Prolog programmers does it take to change a lightbulb? false.
-------------------------------------
Enter your input:  Hows the weather in Macau
Assistant:
The current temperature is 12.85 degrees Celsius.
The weather condition is scattered clouds.
-------------------------------------
```

Enter your input:

# Closing Jupyter Notebook