

2019

Python dictionary

Natalia Novak

Bronx Community College, City University of New York

How does access to this work benefit you? Let us know!

Follow this and additional works at: https://academicworks.cuny.edu/bx_oers

 Part of the [Other Computer Sciences Commons](#)

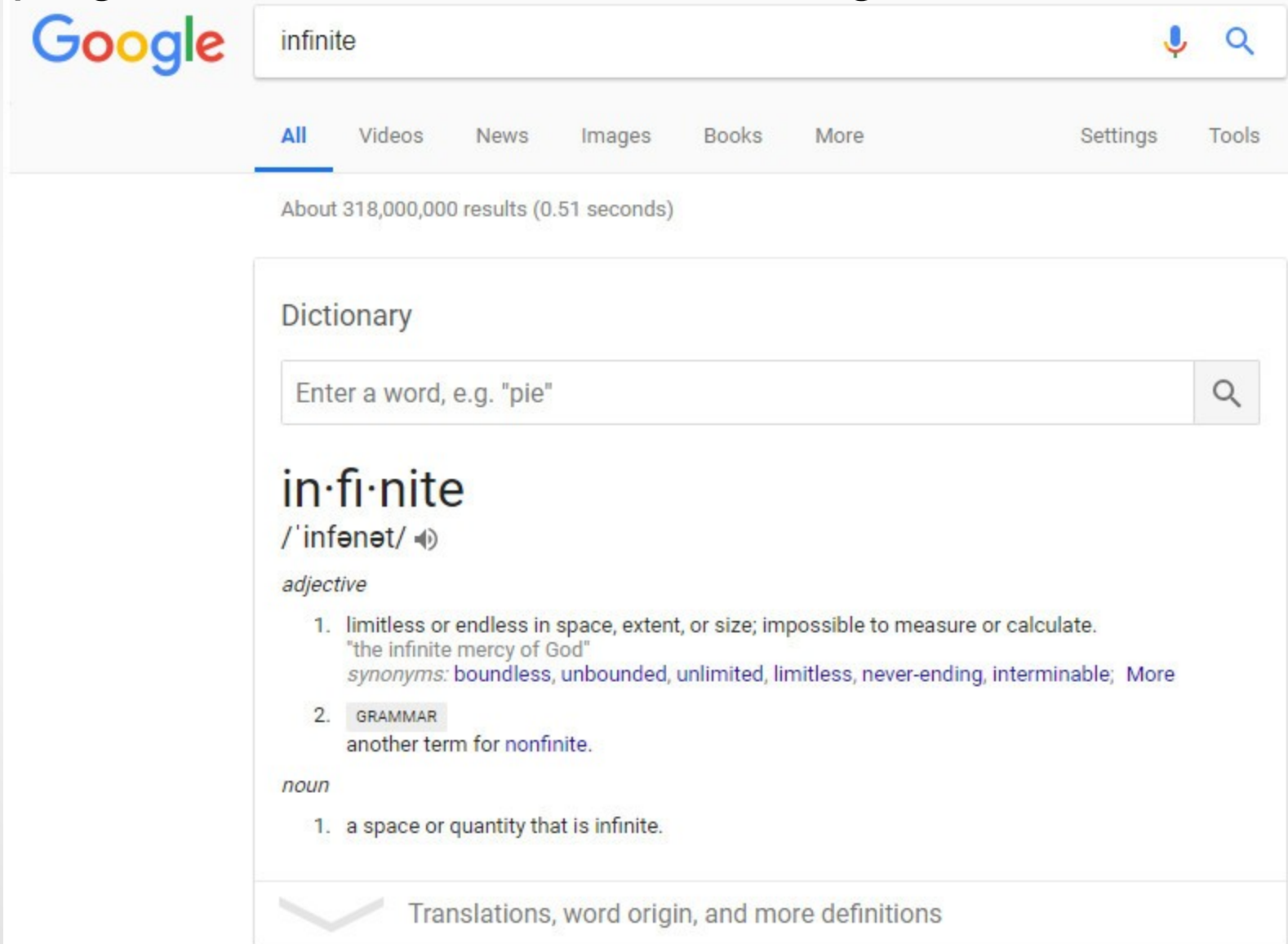
Recommended Citation

Novak, Natalia, "Python dictionary" (2019). *CUNY Academic Works*.
https://academicworks.cuny.edu/bx_oers/26

This Lecture or Presentation is brought to you for free and open access by the Bronx Community College at CUNY Academic Works. It has been accepted for inclusion in Open Educational Resources by an authorized administrator of CUNY Academic Works. For more information, please contact AcademicWorks@cuny.edu.

Dictionary basics

Consider typing the word “infinite” in the Google search:



The screenshot shows a Google search interface. The search bar contains the word "infinite". Below the search bar, the "All" tab is selected. The search results show "About 318,000,000 results (0.51 seconds)". A dictionary widget is displayed, featuring a search input field with the placeholder text "Enter a word, e.g. 'pie'". Below the input field, the word "in·fi·nite" is shown with its phonetic transcription "/ 'ɪnfəneɪt /" and a speaker icon. The word is identified as an "adjective". The first definition is "limitless or endless in space, extent, or size; impossible to measure or calculate." followed by the example "the infinite mercy of God" and a list of synonyms: "boundless, unbounded, unlimited, limitless, never-ending, interminable; More". The second definition is under a "GRAMMAR" tab and states "another term for nonfinite." Below the adjective section, the word is identified as a "noun" with the definition "a space or quantity that is infinite." At the bottom of the dictionary widget, there is a chevron icon and the text "Translations, word origin, and more definitions".

Dictionary basics

A *dictionary* is a Python *container* used to describe associative relationships.

A *dictionary* is represented by the **dict** object type.

A *dictionary* *associates* (or "*maps*") *keys* with *values*.

A *key* is a term that can be located in a *dictionary*, such as the word "infinite" in the Google search.

A *value* describes some data associated with a *key*, such as a definition.

A *key* can be any immutable type, such as a number, string, or tuple; a *value* can be any type.

Dictionary basics

A **dict** object is created using curly braces `{ }` to surround the **key:value** pairs that comprise the dictionary contents.

Example:

```
myDict = {  
    "street address": "2155 University Avenue",  
    "city": "Bronx",  
    "state": "New York",  
    "zip code": 10453,  
    "phone": "(718) 289-5100",  
    "admissions": "(718) 289-5895"}  
}
```

Dictionary basics

Dictionaries are typically used in place of lists when an associative relationship exists.

Example: If a program contains a collection of anonymous student test scores, those scores should be stored in a list. However, if each score is associated with a student name, a dictionary could be used to associate student names to their score.

Dictionary basics: in-class work

I have 5 students: Cute Princess, Fairy Queen, Evil Don, Fussy Cat, and Lazy Daisy.

I also have the record of 4 of their test scores.

Let's create a dictionary `students`, where student's IDs will serve as *keys*, and the *value* will be a *list* with five elements/members: student's name, and 4 test scores.

Name	ID	Test 1	Test 2	Test 3	Test 4
Cute Princess	846563	89	67	98	100
Fairy Queen	736542	76	56	83	99
Evil Don	287563	52	81	79	27
Fussy Cat	294512	27	38	100	75
Lazy Daisy	975321	88	99	66	77

Download and save file `Dict1.py` from our web-site

Dictionary basics: in-class work

students dictionary we got:

keys		0	1	2	3	4
846563	→	"Cute Princess"	89	67	98	100
736542	→	"Fairy Queen"	76	56	83	99
287563	→	"Evil Don"	52	81	79	27
294512	→	"Fussy Cat"	27	38	100	75
975321	→	"Lazy Daisy"	88	99	66	77

Dictionary basics: in-class work

Now, let's print some information: put the following lines into *Dict1.py*:

```
print(students[975321])  
print(students[846563])
```

See what happens!

keys		0	1	2	3	4
846563	→	"Cute Princess"	89	67	98	100
736542	→	"Fairy Queen"	76	56	83	99
287563	→	"Evil Don"	52	81	79	27
294512	→	"Fussy Cat"	27	38	100	75
975321	→	"Lazy Daisy"	88	99	66	77

Dictionary basics: in-class work

Let's now calculate Lazy Daisy's average test score: add the following lines of code into *Dict1.py*

```
s = students[975321]
averageTestScore = (s[1]+s[2]+s[3]+s[4])/4
print("Lazy Daisy average test score is",
      averageTestScore)
```

keys		0	1	2	3	4
846563	→	"Cute Princess"	89	67	98	100
736542	→	"Fairy Queen"	76	56	83	99
287563	→	"Evil Don"	52	81	79	27
294512	→	"Fussy Cat"	27	38	100	75
975321	→	"Lazy Daisy"	88	99	66	77

s →

s[0] *s*[1] *s*[2] *s*[3] *s*[4]

Dictionary basics: in-class work

Now, let's add one more record and display the dictionary:

Name	ID	Test 1	Test 2	Test 3	Test 4
"Glad Lad"	625342	98	76	48	80

By adding the following line in *Dict1.py*:

```
students[625342] = ["Glad Lad", 98, 76, 48, 80]  
print(students)
```

keys		0	1	2	3	4
846563	→	"Cute Princess"	89	67	98	100
736542	→	"Fairy Queen"	76	56	83	99
287563	→	"Evil Don"	52	81	79	27
294512	→	"Fussy Cat"	27	38	100	75
975321	→	"Lazy Daisy"	88	99	66	77
625342	→	"Glad Lad"	98	76	48	80

Dictionary basics: in-class work

Now, let's delete the record about Fussy Cat from the dictionary
By adding the following line in *Dict1.py*:

```
del students[294512]  
print(students)
```

keys		0	1	2	3	4
846563	→	"Cute Princess"	89	67	98	100
736542	→	"Fairy Queen"	76	56	83	99
287563	→	"Evil Don"	52	81	79	27
294512	→	"Fussy Cat"	27	38	100	75
975321	→	"Lazy Daisy"	88	99	66	77
625342	→	"Glad Lad"	98	76	48	80

Dictionary basics: *in-class work*

Review all operations we did in the file *Dict1.py* with dictionary *students* before proceeding to the handout.

In class activity: follow the handout

This OER material was produced as a result of the CS04ALL CUNY OER project.



This work is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 4.0 License.