

Forum:[Hack Forums](#)

Thread: [Python Exercise](#)

Streax - July 7, 2019, 4:24 pm

Anyone got a good website or could post a python exercise?

lolwtf - July 7, 2019, 5:35 pm

Here is a fairly large .svg:

<https://svgshare.com/i/Dxk.svg>

Your objective is to remove the left-hand side of the .svg, so it looks like this:

<https://i.imgur.com/r7XvAic.png>

Your output should be valid .svg. I was able to do this in 20 minutes, so I expect you to do this within 6.6 hours because I'm a 20x rockstar developer.

If you share correct code ill share mine.

Allelujah - July 7, 2019, 9:20 pm

reddit has some good stuff too.
r/python

Kallu609 - July 8, 2019, 5:12 am

[LeetCode - The World's Leading Online Programming Learning Platform](#)

Has tons of different exercises and you can use multiple different languages.

FN-2187 - July 9, 2019, 4:18 am

I have been using [Hacker Rank](#). It's has a lot of different types of exercises that you can switch to python. It will run your code and test it against pre generated samples.

Perlo - July 19, 2019, 1:36 pm

One I used during college was: <https://codingbat.com/python>. They have a nice range of exercises filtered per category.

michalmonday - July 19, 2019, 7:04 pm

(Jul 9, 2019 04:18 AM)FN-2187 Wrote: [\[CITE\]](#) I have been using [Hacker Rank](#). It's has a lot of different types of exercises that you can switch to python. It will run your code and test it against pre generated samples.

+1

It's really useful and well designed. Each submission/task has its' own "Discussion" section where you can check how other people solved the same problem to learn from others on how to write cleaner or more efficient code.

For each submission it gives you some points depending on difficulty, the points are not worth anything but it keeps you going:P

There's also sections to solve problems about algorithms, data structures, maths, regex, sql,

databases and AI. In these sections most of the tasks include parts of pre-written code, there's an option to select a programming language from the long list, so this way you can see how the same thing is implemented in different languages. Here's an example:

[Apple and Orange | HackerRank](#)

Provided python code:

Code `#!/bin/python3`

```
import math
import os
import random
import re
import sys
```

```
# Complete the countApplesAndOranges function below.
def countApplesAndOranges(s, t, a, b, apples, oranges):
```

```
if __name__ == '__main__':
    st = input().split()

    s = int(st[0])

    t = int(st[1])

    ab = input().split()

    a = int(ab[0])

    b = int(ab[1])

    mn = input().split()

    m = int(mn[0])

    n = int(mn[1])

    apples = list(map(int, input().rstrip().split()))

    oranges = list(map(int, input().rstrip().split()))

    countApplesAndOranges(s, t, a, b, apples, oranges)
```

And the same thing provided in C after selecting it from the list:

```
Code #include <assert.h>
#include <limits.h>
#include <math.h>
#include <stdbool.h>
#include <stddef.h>
#include <stdint.h>
#include <stdio.h>
#include <stdlib.h>
```

```
#include <string.h>
```

```
char* readline();  
char** split_string(char*);
```

```
// Complete the countApplesAndOranges function below.
```

```
void countApplesAndOranges(int s, int t, int a, int b, int apples_count, int* apples, int  
oranges_count, int* oranges) {
```

```
}
```

```
int main()
```

```
{
```

```
    char** st = split_string(readline());
```

```
    char* s_endptr;
```

```
    char* s_str = st[0];
```

```
    int s = strtol(s_str, &s_endptr, 10);
```

```
    if (s_endptr == s_str || *s_endptr != '\0') { exit(EXIT_FAILURE); }
```

```
    char* t_endptr;
```

```
    char* t_str = st[1];
```

```
    int t = strtol(t_str, &t_endptr, 10);
```

```
    if (t_endptr == t_str || *t_endptr != '\0') { exit(EXIT_FAILURE); }
```

```
    char** ab = split_string(readline());
```

```
    char* a_endptr;
```

```
    char* a_str = ab[0];
```

```
    int a = strtol(a_str, &a_endptr, 10);
```

```
    if (a_endptr == a_str || *a_endptr != '\0') { exit(EXIT_FAILURE); }
```

```
    char* b_endptr;
```

```
    char* b_str = ab[1];
```

```
    int b = strtol(b_str, &b_endptr, 10);
```

```
    if (b_endptr == b_str || *b_endptr != '\0') { exit(EXIT_FAILURE); }
```

```
    char** mn = split_string(readline());
```

```
    char* m_endptr;
```

```
    char* m_str = mn[0];
```

```
    int m = strtol(m_str, &m_endptr, 10);
```

```
    if (m_endptr == m_str || *m_endptr != '\0') { exit(EXIT_FAILURE); }
```

```
    char* n_endptr;
```

```
    char* n_str = mn[1];
```

```
    int n = strtol(n_str, &n_endptr, 10);
```

```

if (n_endptr == n_str || *n_endptr != '\0') { exit(EXIT_FAILURE); }

char** apples_temp = split_string(readline());

int* apples = malloc(m * sizeof(int));

for (int i = 0; i < m; i++) {
    char* apples_item_endptr;
    char* apples_item_str = *(apples_temp + i);
    int apples_item = strtol(apples_item_str, &apples_item_endptr, 10);

    if (apples_item_endptr == apples_item_str || *apples_item_endptr != '\0') {
exit(EXIT_FAILURE); }

    *(apples + i) = apples_item;
}

int apples_count = m;

char** oranges_temp = split_string(readline());

int* oranges = malloc(n * sizeof(int));

for (int i = 0; i < n; i++) {
    char* oranges_item_endptr;
    char* oranges_item_str = *(oranges_temp + i);
    int oranges_item = strtol(oranges_item_str, &oranges_item_endptr, 10);

    if (oranges_item_endptr == oranges_item_str || *oranges_item_endptr != '\0') {
exit(EXIT_FAILURE); }

    *(oranges + i) = oranges_item;
}

int oranges_count = n;

countApplesAndOranges(s, t, a, b, apples_count, apples, oranges_count, oranges);

return 0;
}

char* readline() {
    size_t alloc_length = 1024;
    size_t data_length = 0;
    char* data = malloc(alloc_length);

    while (true) {
        char* cursor = data + data_length;
        char* line = fgets(cursor, alloc_length - data_length, stdin);

        if (!line) { break; }
    }

```

```

data_length += strlen(cursor);

if (data_length < alloc_length - 1 || data[data_length - 1] == '\n') { break; }

size_t new_length = alloc_length << 1;
data = realloc(data, new_length);

if (!data) { break; }

alloc_length = new_length;
}

if (data[data_length - 1] == '\n') {
    data[data_length - 1] = '\0';
}

data = realloc(data, data_length);

return data;
}

char** split_string(char* str) {
    char** splits = NULL;
    char* token = strtok(str, " ");

    int spaces = 0;

    while (token) {
        splits = realloc(splits, sizeof(char*) * ++spaces);
        if (!splits) {
            return splits;
        }

        splits[spaces - 1] = token;

        token = strtok(NULL, " ");
    }

    return splits;
}

```

There's like 15 languages to choose from that list.

Streax - July 19, 2019, 8:23 pm

Will check all of them out for sure!

Thank you good sirs :)

2tonstun - July 28, 2019, 10:36 pm

ooo sweet i needed the same thing

Lindros - July 29, 2019, 2:02 pm

w3schools has some good beginner excersizes. Not quite sure what skill level you are looking for.

vorticalbox - July 31, 2019, 6:34 am

(Jul 19, 2019 01:36 PM)Perlo Wrote: [\[CITE\]](#) One I used during college was: <https://codingbat.com/python>. They have a nice range of exercises filtered per category. This is actually pretty fun :P

Influxes - August 14, 2019, 10:28 am

A fun one is to recreate the game fizzbuzz with python. The basics of the game, is you count up from 1. If the number you are currently at is divisible by 3, the program prints "fizz". If the number is divisible by 5 the program prints "buzz". Now if the number is divisible by both 3 and 5, the program prints "fizzbuzz".

Another fun project to do is to build a cipher. A Caesar Cipher. You can read here on how Caesar Ciphers work:

[Caesar cipher - Wikipedia](#)

Matabufalez - August 16, 2019, 5:24 am

<https://www.freecodecamp.org/> is pretty nice too, it even has job offers.

sid071 - August 26, 2019, 4:30 am

(Jul 7, 2019 04:24 PM)Streax Wrote: [\[CITE\]](#) Anyone got a good website or could post a python exercise?

I have some 15 solid practice on my website. 'www.masterpython.me'

(Aug 26, 2019 04:30 AM)sid071 Wrote: [\[CITE\]](#) I have some 15 solid practice on my website.

'www.masterpython.me'

<https://www.masterpython.me>