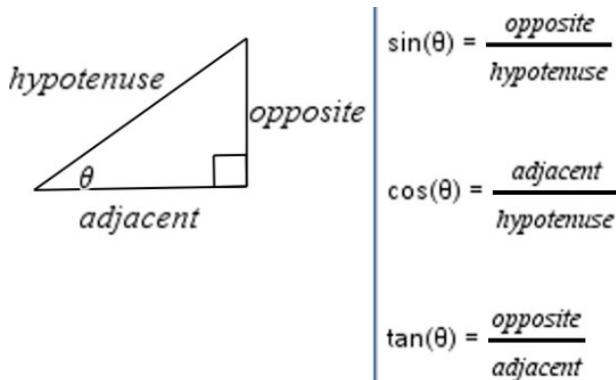


Gettin' Trig-y with Python: Trigonometry Ratios

In this lesson, you will write a program that generates a random sine, cosine or tangent ratio. It will ask the user to enter a different trigonometric ratio based on the given triangle information. It will evaluate the user's input. If the user's input is incorrect, give the correct ratio.



Objectives:

Programming Objectives:

- Use the input function and a variable to collect and store data from a user
- Use the randint function to generate triangle side lengths
- Use the randint function to determine sine, cosine, or tangent prompts
- Use if..elif..else statements to make decisions

Math Objectives:

- Use the Pythagorean Theorem to find missing triangle side lengths
- Use right triangle trigonometry ratios to solve problems

For this project, you will write a program that generates a random sine, cosine or tangent ratio. It will ask the user to enter a different trigonometric ratio based on the given triangle information. It will evaluate the user's input. If the user's input is incorrect, it will give the correct ratio.

Problem 1 Prompt

```

PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running T
>>> from T import *
given cos(theta)= 8 / sqrt(68)
-----
tan(theta)= |
    
```

Problem 2 Prompt

```

PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running TRIG_PRA
>>> from TRIG_PRA import *
given tan(theta)= 8 / sqrt(17)
-----
cos(theta)= |
    
```

```

PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running T
>>> from T import *
given cos(theta)= 8 / sqrt(68)
-----
tan(theta)= 4/10
sorry: 2 / 8
>>> |
    
```

Fns... a A # Tools Editor Files

Incorrect answer entered,
gives the correct answer.

```

PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running TRIG_PRA
>>> from TRIG_PRA import *
given tan(theta)= 8 / sqrt(17)
-----
cos(theta)= sqrt(17)/9
correct
>>> |
    
```

Fns... a A # Tools Editor Files

Correct answer was given.



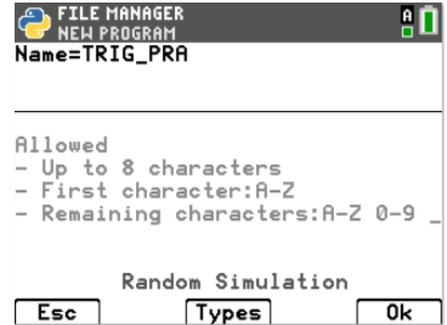
1. Start a new Python project. Name the project **TRIG_PRA**.

To access the underscore, `_`, type [2nd] [(-)].

You do not have to put the underscore in the name of the file. You could name the file TRIGPRA

You are not allowed to use a space in the name of a project. Often, programmers will use an underscore to separate words variables and project names.

This project will utilize the `randint` function to generate random integers. This requires the `random` library. Select **Random Simulation** as the type.



2. Make sure the `random` library was imported.

If you forgot to select **Random Simulation**, you can go to `Fns> Modul> Random> from random import*`

The comments, denoted by the `#` appear in light gray. These lines of code are optional. They do not affect how the program executes. Instead, they are notes to you, the programmer.

Comments are used to describe sections of code. This can be useful when debugging or modifying a program later. They help a programmer remember the purpose for different sections of code. These comments tell you the coder; this library is used for data sharing.



3. Import the `math` library.

`Fns> Modul> Math> from math import *`





Math Explorations with Python

TI-84 PLUS CE PYTHON TECHNOLOGY

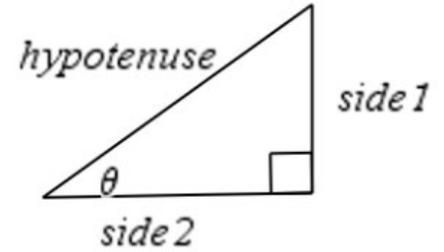
GETTIN' TRIG-Y WITH PYTHON

STUDENT ACTIVITY

- The project will generate two integer sides for a right triangle. It will calculate the missing side.

What is the name of the formula than can be used to find the missing side of a right triangle?

What is the formula that can be used to find a missing side of a triangle given two sides?



- Using the Pythagorean theorem, $(side_1)^2 + (side_2)^2 = hypotenuse^2$, you will create random right triangle side lengths.

To mix up the measurements, 1/3 of the time the sides will be integers, 1/3 of the time side1 and the hypotenuse will be integers and the remaining time side2 and the hypotenuse will be integers.

You will use an **if..elif..else** statement to make find the three options.

```
if randint(1,3) == 1:
    #generate two integer side lengths
    #calculate the hypotenuse
elif randint(1,2) == 1:
    #generate opposite side and hypotenuse integers
    #calculate the adjacent side
else:
    #generate adjacent side and hypotenuse integers
    #calculate the opposite side
```

- Add a comment to document this section of code.

[Tools] → C: Insert #comment below

or

[a A #] → # → [Paste]

#create sides



Add the if statement for the first option

Fns> Ctl> If..elif..else



Math Explorations with Python

TI-84 PLUS CE PYTHON TECHNOLOGY

- The function `randint(1,3)` will generate numbers 1,2 and 3.
The function `randint(1,2)` will generate numbers 1 and 2.

Inside the first if statement, type
`randint(1,3) == 1`

Inside the elif statement type
`randint(1,2) == 1`

Fns> Modul> Random> randint

- Use `randint` to generate integer side lengths from 1 to 10.
In the if statement generate the opposite (`op`) and adjacent (`adj`).

In the elif statement generate the hypotenuse (`h`) and opposite (`op`).
 The hypotenuse needs to be larger than the opposite side.
 Therefore, start the hypotenuse options one more than `op`.

In the else statement generate the hypotenuse (`h`) and the adjacent (`adj`).
 The hypotenuse needs to be larger than the adjacent side.
 Therefore, start the hypotenuse options one more than `adj`.

****Tech Tip**** the [Tools] menu has a copy line and paste line option.
It could be helpful to copy, paste and alter lines.

- Use the Pythagorean theorem,
 $(\text{opposite})^2 + (\text{side}_2)^2 = \text{hypotenuse}^2$
 to find the missing side length for each part of the if..elif..else.

To take the square root, type `sqrt()` or **Fns> Modul> Math> Sqrt**

To square each side you can multiple the side by itself, such as
`adj*adj`
 or use `**2` to square such as
`adj**2`

GETTIN' TRIG-Y WITH PYTHON

STUDENT ACTIVITY

```

EDITOR: TRIG_PRA
PROGRAM LINE 0011
# Random Simulation
from random import *
from math import *

#create sides
if randint(1,3)==1:
**
elif randint(1,2)==1:
**
else:
**
_

```

```

EDITOR: TRIGPRAC
PROGRAM LINE 0014
#create sides
if randint(1,3,)==1:
*op=randint(1,10)
*adj=randint(1,10)
elif randint(1,2)==1:
*op=randint(1,9)
*h=randint(op+1,10)
else:
*adj=randint(1,9)
*h=randint(adj+1,10)

```

```

EDITOR: TRIGPRAC
PROGRAM LINE 0017
*op=randint(1,10)
*adj=randint(1,10)
*h=sqrt(op**2+adj**2)
elif randint(1,2)==1:
*op=randint(1,9)
*h=randint(op+1,10)
*adj=sqrt(h**2-op**2)
else:
*adj=randint(1,9)
*h=randint(adj+1,10)
*op=sqrt(h**2-adj**2)_

```



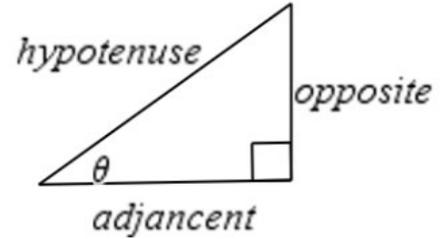
10. Double check your equations.

Did you use:

$$h = \sqrt{op^2 + adj^2}$$

$$op = \sqrt{h^2 - adj^2}$$

$$adj = \sqrt{h^2 - op^2}$$



11. Right now the values for the side lengths are stored as integers.

For display purposes we will need a string copy.

If the length is an integer, we will simply put `str()` around the value and save it under a different name.

For example, in the first if statement you found and stored the value of the adjacent side using the formula:

```
adj=randint(1,10)
```

To make a display string you will type

```
adj1=str(adj)
```

If the length is a number with a square root, it most likely is an irrational number. Instead of displaying the value as an evaluated decimal value, we will get a little fancy. We will put the words "sqrt(" in front of the non-evaluated number.

For example, in the first if statement you found and stored the value of the hypotenuse using the formula:

```
h=sqrt(op**2+adj*2)
```

To make a display value you will type:

```
h1 = "sqrt(" + str(op**2+adj**2) + ")"
```

```

if randint(1,3)==1:
    op=randint(1,10)
    adj=randint(1,10)
    h=sqrt(op**2+adj*2)
    op1=str(op)
    adj1=str(adj)
    h1="sqrt("+str(op**2+adj**2)+")"
elif randint(1,2)==1:
    h=randint(1,10)

```

```

EDITOR: TRIG_PRA
PROGRAM LINE 0015
if randint(1,3)==1:
    op=randint(1,10)
    adj=randint(1,10)
    h=sqrt(op**2+adj**2)
    op1=str(op)
    adj1=str(adj)
    h1="sqrt("+str(op**2+adj**2)+")"
elif randint(1,2)==1:
    op=randint(1,9)
    h=randint(op+1,10)

```



```

op=randint(1,10)
adj=sqrt(h**2-op**2)
h1=str(h)
op1=str(op)
adj1="sqrt("+str(h**2-op**2)+")"
else:
h=randint(1,10)
adj=randint(1,10)
op=sqrt(h**2-op**2)
h1=str(h)
adj1=str(adj)
op1="sqrt("+str(h**2-op**2)+")"

```

****Tech Tip**** the [Tools] menu has a copy line and paste line option. It could be helpful to copy, paste and alter lines

12. Let's check your code so far. In large projects such as this one, programmers execute the code often to check for errors.

Add three print lines to display the three display values. Make sure the print lines are **NOT INDENTED**.

Fns>> I/O> print

```

print(adj1)
print(op1)
print(h1)

```

Execute the program multiple times. **Run**

Each time only one value should have sqrt() displayed.

GETTIN' TRIG-Y WITH PYTHON STUDENT ACTIVITY

```

*adj=sqrt(h**2-op**2)
*op1=str(op)
*h1=str(h)
*adj1="sqrt("+str(h**2-op**2)+")"
else:
*adj=randint(1,9)
*h=randint(adj+1,10)
*op=sqrt(h**2-adj**2)
*adj1=str(adj)
*h1=str(h)
*op1="sqrt("+str(h**2-adj**2)+")"

```

EDITOR: TRIG_PRA
PROGRAM LINE 0030

```

*adj=randint(1,9)
*h=randint(adj+1,10)
*op=sqrt(h**2-adj**2)
*adj1=str(adj)
*h1=str(h)
*op1="sqrt("+str(h**2-adj**2)+")"

print(adj1)
print(op1)
print(h1)

```

Fns... | a A # | Tools | Run | Files

Sample Runs:

PYTHON SHELL

```

>>> # Shell Reinitialized
>>> # Running TRIG_PRA
>>> from TRIG_PRA import *
sqrt(36)
8
10
>>> |

```

Fns... | a A # | Tools | Editor | Files

The first time adj1 is a square root.

```

PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running TRIG_PRA
>>> from TRIG_PRA import *
2
sqrt(60)
8
>>> |
    
```

The second time op1 is a square root.

13. Remove the three print lines.

This app should let you practice three different trig functions.

For example, it might give

the ratio for $\sin(\theta)$ and ask the user to find the ratio for $\tan(\theta)$

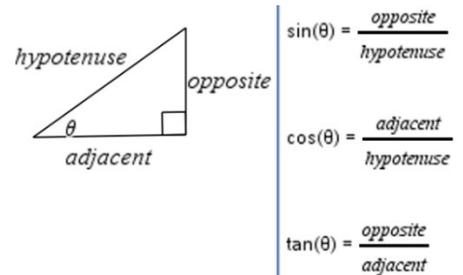
or

the ratio for $\cos(\theta)$ and ask the user to find the ratio for $\sin(\theta)$

There are three choices for the given function and three for the function to find.

given = randint(1,3)

find = randint(1,3)



```

EDITOR: TRIG_PRA
PROGRAM LINE 0030
*adj=randint(1,9)
*h=randint(adj+1,10)
*op=sqrt(h**2-adj**2)
*adj1=str(adj)
*h1=str(h)
*op1="sqrt("+str(h**2-adj**2)+"
)"
given=randint(1,3)
find=randint(1,3)
    
```

14. It is highly likely (1/3) that the variable to find matches the given variable.

To fix this, add a while loop that will continually generate the find until while the find matches the given.

while find == given:

find=randint(1,3)

Fns > Ctl> while

Make sure to use == to check if they match.

```

EDITOR: TRIG_PRA
PROGRAM LINE 0031
*h=randint(adj+1,10)
*op=sqrt(h**2-adj**2)
*adj1=str(adj)
*h1=str(h)
*op1="sqrt("+str(h**2-adj**2)+"
)"
given=randint(1,3)
find=randint(1,3)
while find==given:
*find=randint(1,3)_
    
```



Math Explorations with Python

TI-84 PLUS CE PYTHON TECHNOLOGY

15. If the given is a 1, display the ratio for $\sin(\theta)$.
 If the given is a 2, display the ratio for $\cos(\theta)$.
 If the given is a 3, display the ratio for $\tan(\theta)$.
 Outside the if statement print "-----"

```
if given == 1:
    print("Given: sin(theta) =",op1,"/",h1)
elif given == 2:
    print("Given: cos(theta) =",adj1,"/",h1)
else:
    print("Given: tan(theta) =",op1,"/",adj1)
print("-----")
```

16. A similar if..elif..else statement will print the ratio to find, retrieve the user's result **AND** determine if the user is correct.

To start let's display the question.

```
if find == 1:
    ans=eval(input("sin(theta) = "))
elif find == 2:
    ans=eval(input("cos(theta) = "))
else:
    ans=eval(input("tan(theta) = "))
```

17. Now, compare the entered answer to the correct answer.

For example, if $find = 1$, compare the answer to $\sin(\theta)$.
 We need to know if $ans == op/h$.
 Because the answer is stored as a float (decimal), and op/h evaluates to a decimal, it is possible to have a round-off error when comparing. To fix this, we will check to see if the difference between the answer and op/h is less than 0.00001.

```
if abs(ans - op/h) < 0.00001:
    print("correct")
else:
    print("Sorry",op1,"/",h1)
```

GETTIN' TRIG-Y WITH PYTHON

STUDENT ACTIVITY

```
EDITOR: TRIG_PRA
PROGRAM LINE 0032
while find==given:
    find=randint(1,3)

if given==1:
    print("given sin(theta)=",op1,
        "/",h1)
elif given==2:
    print("given cos(theta)=",adj1,
        "/",h1)
else:
    print("given tan(theta)=",op1,
        "/",adj1)
print("-----")
```

```
EDITOR: T
PROGRAM LINE 0046
print("-----")

if find==1:
    ans=eval(input("sin(theta)= "))
elif find==2:
    ans=eval(input("cos(theta)= "))
else:
    ans=eval(input("tan(theta)= "))
```

```
EDITOR: TRIG_PRA
PROGRAM LINE 0049
if find==1:
    ans=eval(input("sin(theta)= "))
    if abs(ans-op/h)<0.00001:
        print("correct")
    else:
        print("sorry:",op1,"/",h1)
elif find==2:
    ans=eval(input("cos(theta)= "))
    if abs(ans-adj/h)<0.00001:
        print("correct")
    else:
        print("sorry:",adj1,"/",h1)
else:
    ans=eval(input("tan(theta)= "))
    if abs(ans-op/adj)<0.00001:
        print("correct")
    else:
        print("sorry:",op1,"/",adj1)
```

18. Execute your program. Run

Sample 1

$$\cos(\theta) = \frac{8}{\sqrt{68}} = \frac{\text{adjacent}}{\text{hypotenuse}}$$

$$\text{opposite}^2 + \text{adjacent}^2 = \text{hypotenuse}^2$$

$$\text{opposite}^2 + 8^2 = \sqrt{68}^2$$

$$4 = \text{opposite}^2$$

$$\text{opposite} = \sqrt{4} = 2$$

$$\tan(\theta) = \frac{\text{opposite}}{\text{adjacent}} = \frac{2}{8}$$

Sample 2

$$\tan(\theta) = \frac{8}{\sqrt{17}} = \frac{\text{opposite}}{\text{adjacent}}$$

$$\text{opposite}^2 + \text{adjacent}^2 = \text{hypotenuse}^2$$

$$8^2 + \sqrt{17}^2 = \text{hypotenuse}^2$$

$$\text{hypotenuse}^2 = 64 + 17$$

$$\text{hypotenuse} = \sqrt{81} = 9$$

$$\cos(\theta) = \frac{\text{adjacent}}{\text{hypotenuse}} = \frac{\sqrt{17}}{9}$$

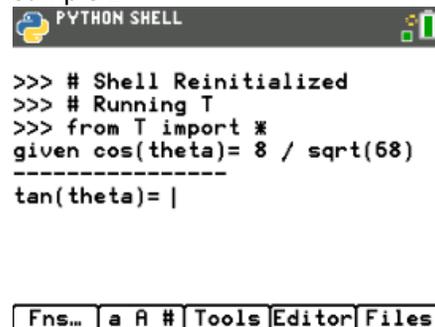
19. Now that you have a trigonometry practice app, can you answer 5 questions in a row correctly?

Can you get 10 in a row correct? Go on, Get Trig-y with it!

GETTIN' TRIG-Y WITH PYTHON

STUDENT ACTIVITY

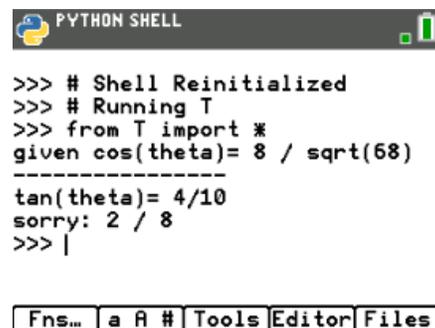
Sample 1



```

PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running T
>>> from T import *
given cos(theta)= 8 / sqrt(68)
-----
tan(theta)= |
    
```

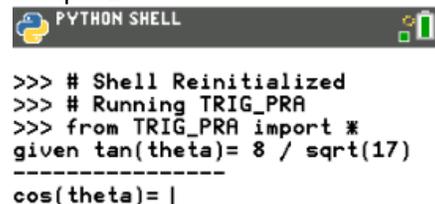
Entered 4/10 incorrect answer



```

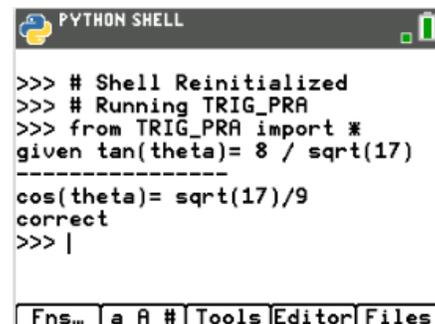
PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running T
>>> from T import *
given cos(theta)= 8 / sqrt(68)
-----
tan(theta)= 4/10
sorry: 2 / 8
>>> |
    
```

Sample 2



```

PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running TRIG_PRA
>>> from TRIG_PRA import *
given tan(theta)= 8 / sqrt(17)
-----
cos(theta)= |
    
```



```

PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running TRIG_PRA
>>> from TRIG_PRA import *
given tan(theta)= 8 / sqrt(17)
-----
cos(theta)= sqrt(17)/9
correct
>>> |
    
```