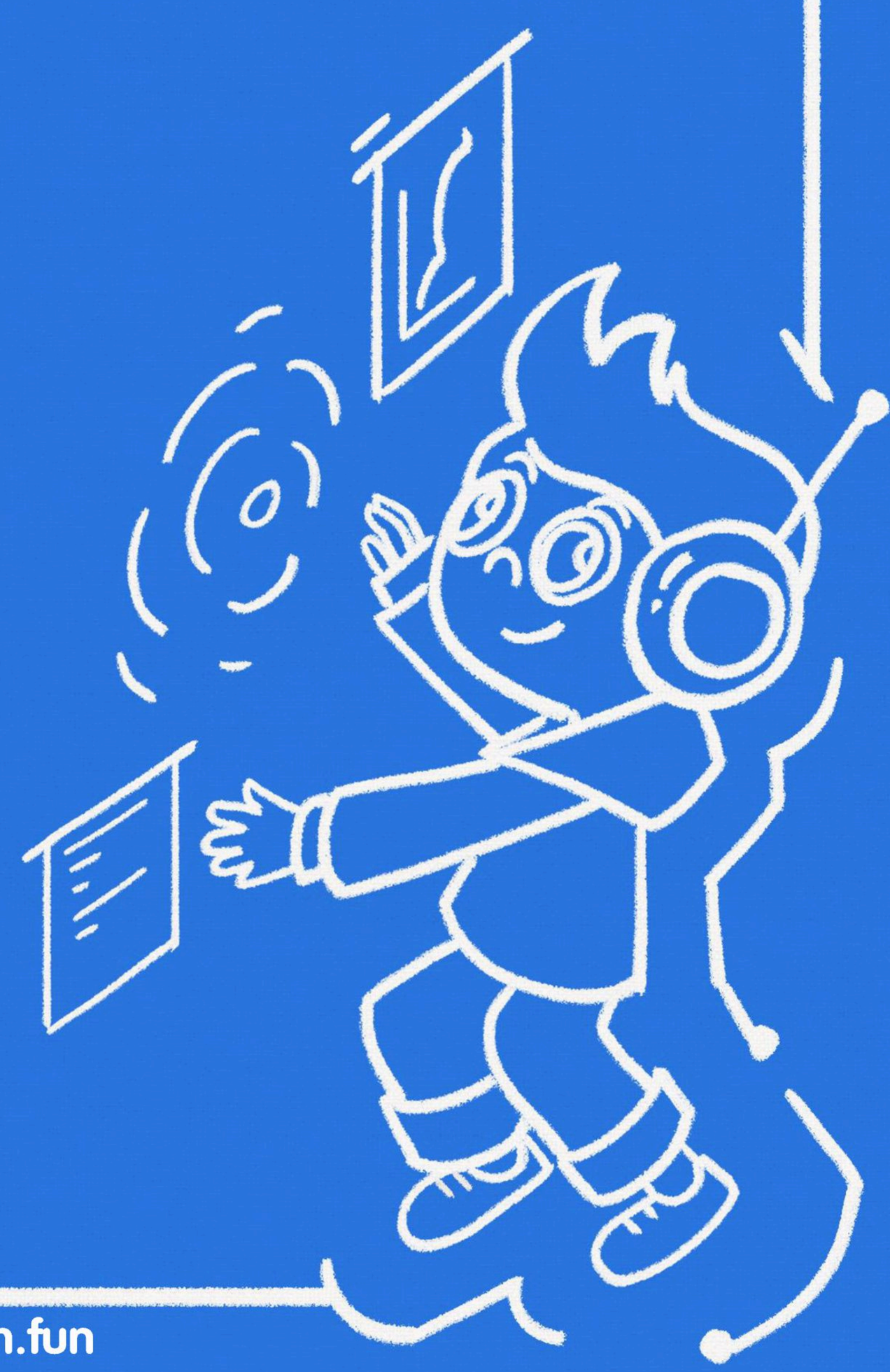
知识树



1.探索新知

1.1 初始化设置

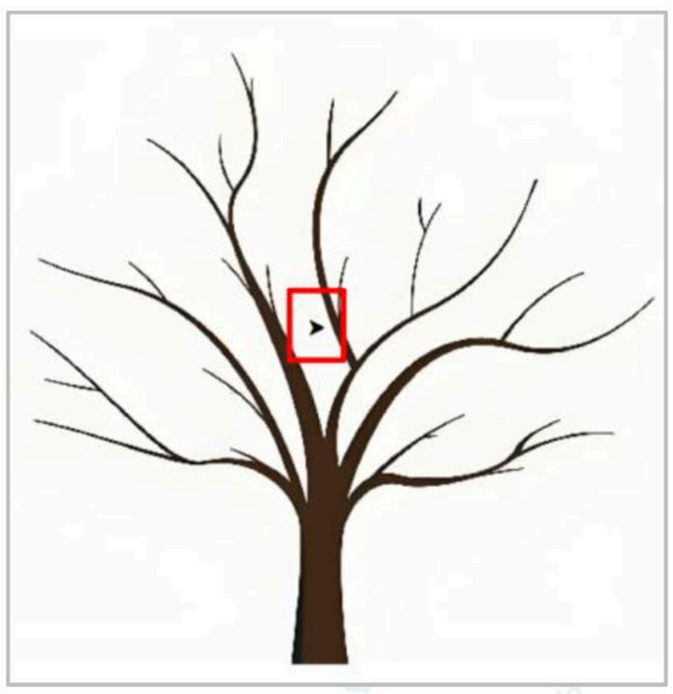
1. 设置屏幕

```
# 设置屏幕
turtle.bgpic("树干.png")
```


```
# 停止画笔绘制，但绘图窗体不关闭
turtle.done()
```

2. 创建画笔

```
# 创建画笔
pen = turtle.Turtle()
```



3. 隐藏画笔

```
# 隐藏画笔
pen.hideturtle()
```

1.探索新知

1.1 初始化设置

4. 创建颜色列表

```
# 创建颜色列表
colors = ["red", "orange",  "green", "blue", "purple", "pink", "black"]
```

花瓣有多种颜色，我们可以把多种颜色存到一个列表中，画花花时随机选用

任务一新增代码如下：

```
import turtle

# 设置屏幕
turtle.bgpic("树干.png")
# 创建画笔
pen = turtle.Turtle()
# 隐藏画笔
pen.hideturtle()
# 创建颜色列表
colors = ["red","orange",  "green", "blue", "purple", "pink", "black"]

# 停止画笔绘制，但绘图窗体不关闭
turtle.done()
```

1.2 点击画花花

鼠标点击时，会在点击的位置画出一朵花。
那如何知道鼠标点击了哪里？

turtle.onscreenclick(fun)

检测鼠标点击事件。
当鼠标在屏幕上点击时，将调用fun函数，并将点击时鼠标的x和y坐标作为参数传递给fun。

例如：

```
import turtle
def onClick(x,y):
    print(x, y)
turtle.onscreenclick(onClick)
turtle.done()
```

1. 检测鼠标点击事件

接下来我们完成【画花花】的函数，想想要分几步呢？

```
# 画花花
def draw_flower(x, y):
    ...

# 窗口点击事件监听
turtle.onscreenclick(draw_flower)
```

2. 画花花

```
# 画花花
def draw_flower(x, y):
    # <1. 抬笔，画笔移到鼠标位置落笔>
    # <2. 绘制彩色花瓣>
    # <3. 绘制黄色花蕊>
```

 # <1. 抬笔，画笔移到鼠标位置落笔>

```
# 移动画笔但不绘制
pen.penup()
pen.goto(x, y)
pen.pendown()
```

 # <2. 绘制彩色花瓣>

① 从颜色列表随机选择一种颜色

别忘了导入random库

```
# <2. 绘制彩色花瓣>
# 随机选择花瓣颜色
pen_color = random.choice(colors)
```

② 设置画笔和填充色

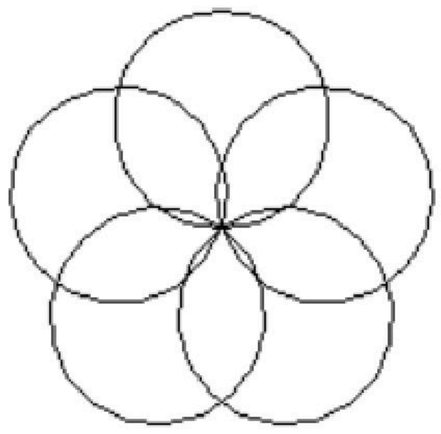
```
# <2. 绘制彩色花瓣>
# 随机选择花瓣颜色
pen_color = random.choice(colors)
# 设置画笔和填充颜色
pen.color(pen_color)
```

1.2 点击画花花

<2. 绘制彩色花瓣>

③ 绘制花瓣轮廓

```
# 画出一个小花
for i in range(5):
    pen.circle(10)
    pen.right(72)
```



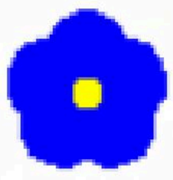
④ 花瓣填色

```
pen.begin_fill() # 开始填充颜色
for i in range(5):
    pen.circle(10)
    pen.right(72)
pen.end_fill()   # 填充完成
```

<3. 绘制黄色花蕊>

黄色花蕊其实是黄色的实心圆。
你能说出几种绘制实心圆的方法？

```
pen.color("yellow")
pen.dot(8)
```



花蕊在中心

```
pen.color("yellow")
pen.begin_fill()
pen.circle(4)
pen.end_fill()
```

花蕊不在中心

1.2 点击画花花

任务二新增代码如下：

```
import random

# 画花花
def draw_flower(x, y):
    # 移动画笔但不绘制
    pen.penup()
    pen.goto(x, y)
    pen.pendown()
    pen_color = random.choice(colors)  # 随机选择花瓣颜色
    pen.color(pen_color)   # 设置画笔和填充颜色
    pen.begin_fill()   # 开始填充颜色
    # 画出一个小花
    for i in range(5):
        pen.circle(10)
        pen.right(72)
    pen.end_fill()     # 填充完成
    # 绘制黄色花蕊
    pen.color("yellow")
    pen.dot(8)
# 窗口点击事件监听
turtle.onscreenclick(draw_flower)
```

1.3 效果优化

`turtle.tracer(0)`:

加速图形绘制。关闭绘图窗口的绘图动画效果，立即显示绘图结果。

完整代码

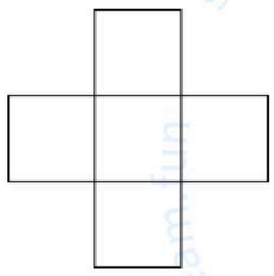
```
import turtle
import random

turtle.bgpic("树干.png")  # 设置屏幕
pen = turtle.Turtle()  # 创建画笔
pen.hideturtle()  # 隐藏画笔
# 创建颜色列表
colors = ["red", "orange",  "green", "blue", "purple", "pink",
"black"]

# 画花花
def draw_flower(x, y):
    # 移动画笔但不绘制
    pen.penup()
    pen.goto(x, y)
    pen.pendown()
    # 随机选择花瓣颜色
    pen_color = random.choice(colors)
    # 设置画笔和填充颜色
    pen.color(pen_color)
    # 开始填充颜色
    pen.begin_fill()
    # 画出一个小花
    for i in range(5):
        pen.circle(10)
        pen.right(72)
    # 填充完成
    pen.end_fill()
    # 绘制黄色花蕊
    pen.color("yellow")
    pen.dot(8)

# 窗口点击事件监听
turtle.onscreenclick(draw_flower)
# 将窗口刷新关闭
turtle.tracer(0)
# 停止画笔绘制，但绘图窗体不关闭
turtle.done()
```

2.强化练习

1．运行下方代码段，输出的结果是（　　）

```
import turtle
def draw():
    turtle.fd(80)
    turtle.left(90)
    turtle.fd(40)
    turtle.left(90)
    turtle.fd(80)
turtle.hideturtle()
for i in range(4):
    turtle.left(90)
    draw()
```

A.



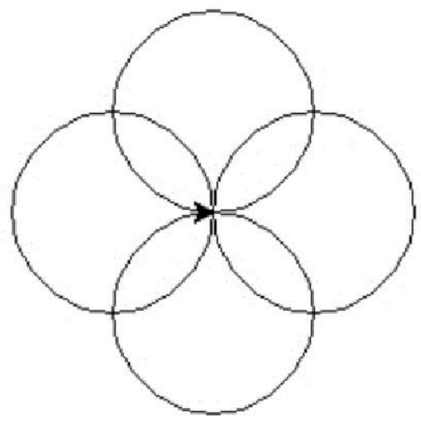
B.

C.

D.

2．为画出如下所示图形，下面Python代码横线处应填入（　　）

```
import turtle
for i in range(4):
    turtle.circle(50)
    turtle.left(_____________)
```



A. 360

B. 180

C. 90

D. 45

3．下列turtle工具箱的语句，哪一项绘制出的形状与其他项不同？（　　）

A. turtle.forward(100)

B. turtle.backward(-100)

C. turtle.goto(100,0)

D. turtle.dot(100)

2.强化练习

4．下面哪个命令是用来检测鼠标点击事件的？（　　）

A. turtle.onscreenclick(fun)

B. turtle.penup()

C. turtle.forward(100)

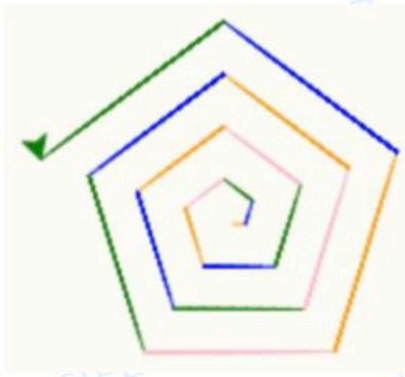
D. turtle.circle(50)

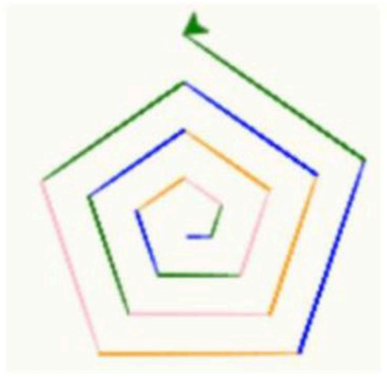
5．阅读下面代码，海龟绘制结束后将会显示的画面是（　　）

```
import turtle
colors =['pink','orange','blue','green']
for i in range(10,100,5):
    turtle.color(colors[i %len(colors)])
    turtle.forward(i)
    turtle.left(72)
```

A. 

B. 

C. 

D. 

3.术语箱

draw	画画
flower	花，花朵
click	点击，单击

4.课后挑战

点点星光

要求：

1. 导入背景“黑夜.png”
2. 初始化设置
3. 点击画星星
4. 效果优化