

# در پایتون: Turtle راهنمای جامع کتابخانه پروژه برتر برای ترسیم و خلاقیت

## مقدمه

در پایتون یکی از جذابترین و در عین حال سادهترین ابزارها برای ورود به دنیای Turtle کتابخانه (Turtle Graphics) برنامه‌نویسی و گرافیک است. این کتابخانه که بر اساس مفهوم گرافیک لاکپشتی توسعه یافته، به برنامه‌نویسان، به ویژه مبتدیان، این امکان را می‌دهد که با استفاده از دستورات متنی ساده، اشکال هندسی، الگوهای پیچیده و حتی بازی‌های کوچک را به صورت بصری ترسیم کنند.

تصور کنید یک لاکپشت کوچک با یک قلم در اختیار دارید که روی یک صفحه سفید قرار گرفته است. هر دستوری که شما به این لاکپشت می‌دهید (مانند جلو رفتن، چرخیدن به چپ یا راست)، اثری از قلم آن بر روی صفحه باقی می‌گذارد. این روش بصری، یادگیری مفاهیم پایه‌ای برنامه‌نویسی مانند حلقه‌ها، شرطها و توابع را بسیار ملموس و لذت‌بخش می‌کند.

در این راهنمای جامع، ما نه تنها با اصول کار با این کتابخانه آشنا می‌شویم، بلکه ۲۲ نمونه کد برتر و کاربردی را بررسی خواهیم کرد که قدرت و انعطاف‌پذیری این ابزار را به نمایش می‌گذارند.

## آشنایی با مفاهیم و دستورات پایه

کنید (Import) تنها کافی است آن را وارد، Turtle برای شروع کار با کتابخانه

```
import turtle
```

ایجاد (Turtle) و یک شیء لاکپشت (Screen) پس از وارد کردن، معمولاً یک صفحه نمایش می‌کنیم.

دستور	توضیح
<code>turtle.Screen()</code>	ایجاد پنجره‌ای که ترسیم در آن انجام می‌شود.
<code>turtle.Turtle()</code>	ایجاد شیء لاک‌پشت (قلم) که دستورات را اجرا می‌کند.
<code>t.forward(فاصله)</code>	لاک‌پشت به اندازه <code>فاصله</code> به جلو حرکت می‌کند.
<code>t.backward(فاصله)</code>	لاک‌پشت به اندازه <code>فاصله</code> به عقب حرکت می‌کند.
<code>t.right(زاویه)</code>	لاک‌پشت به اندازه <code>زاویه</code> به راست می‌چرخد.
<code>t.left(زاویه)</code>	لاک‌پشت به اندازه <code>زاویه</code> به چپ می‌چرخد.
<code>t.penup()</code>	قلم را بالا می‌برد (حرکت بدون ترسیم).
<code>t.pendown()</code>	قلم را پایین می‌آورد (حرکت همراه با ترسیم).
<code>t.speed(سرعت)</code>	سرعت ترسیم را تنظیم می‌کند (از ۱ تا ۱۰، و ۰ برای سریع‌ترین حالت).
<code>t.color(رنگ)</code>	رنگ قلم را تغییر می‌دهد (مانند <code>"red"</code> , <code>"blue"</code> , <code>"#FF0000"</code> ).
<code>turtle.done()</code>	پنجره ترسیم را باز نگه می‌دارد تا کاربر آن را ببندد.

## Turtle پروژه برتر با کتابخانه ۲۲

این پروژه در چهار دسته اصلی، از ساده تا پیشرفته، دسته‌بندی شده‌اند تا شما را در مسیر یادگیری همراهی کنند.

### الف) اشکال هندسی پایه و الگوهای ساده (پروژه‌های ۱ تا ۶)

#### ۱. ترسیم مربع

```
import turtle
t = turtle.Turtle()
for _ in range(4):
    t.forward(100)
    t.right(90)
turtle.done()
```

## ۲. ترسیم ستاره پنج‌پر

```
import turtle
t = turtle.Turtle()
for _ in range(5):
    t.forward(100)
    t.right(144)
turtle.done()
```

## ۳. ترسیم دایره و نقطه

```
import turtle
t = turtle.Turtle()
t.circle(50)
t.penup()
t.goto(0, -70)
t.pendown()
t.dot(40, "red")
turtle.done()
```

## ۴. ترسیم مارپیچ رنگین‌کمانی

```
import turtle
import colorsys
t = turtle.Turtle()
t.speed(0)
for i in range(100):
    color = colorsys.hsv_to_rgb(i/100, 1.0, 1.0)
    t.pencolor(color)
    t.forward(i * 2)
    t.right(91)
turtle.done()
```

## ۵. ترسیم چندضلعی‌های متوالی

```
import turtle
t = turtle.Turtle()
def draw_poly(sides, length):
    for _ in range(sides):
        t.forward(length)
        t.right(360/sides)
for s in range(3, 9):
    draw_poly(s, 50)
turtle.done()
```

## ۶. ترسیم پرچم ایران

```
import turtle
t = turtle.Turtle()
def draw_rect(color, x, y, w, h):
    t.penup()
    t.goto(x, y)
    t.pendown()
    t.color(color)
    t.begin_fill()
    for _ in range(2):
        t.forward(w)
        t.right(90)
        t.forward(h)
        t.right(90)
    t.end_fill()
draw_rect("green", -150, 100, 300, 60)
draw_rect("white", -150, 40, 300, 60)
draw_rect("red", -150, -20, 300, 60)
turtle.done()
```

## ب) الگوهای هنری و پیچیده (پروژه‌های ۷ تا ۱۲)

### ۷. حلقه‌های المپیک

```
import turtle
t = turtle.Turtle()
t.width(5)
rings = [("blue", -110, 0), ("black", 0, 0), ("red", 110, 0), ("yellow",
-55, -50), ("green", 55, -50)]
for color, x, y in rings:
    t.penup()
    t.goto(x, y)
    t.pendown()
    t.color(color)
    t.circle(50)
turtle.done()
```

### ۸. الگوی ماندالا (Mandala)

```
import turtle
t = turtle.Turtle()
t.speed(0)
for i in range(36):
    t.circle(100)
    t.right(10)
turtle.done()
```

### ۹. ترسیم گل و برگ

```
import turtle
t = turtle.Turtle()
t.speed(0)
t.color("red", "yellow")
t.begin_fill()
for _ in range(36):
    t.forward(200)
    t.left(170)
t.end_fill()
turtle.done()
```

## ۱۰. ترسیم صفحه شطرنج

```
import turtle
t = turtle.Turtle()
t.speed(0)
for i in range(8):
    for j in range(8):
        t.penup()
        t.goto(j*30, i*30)
        t.pendown()
        if (i+j) % 2 == 0: t.begin_fill()
        for _ in range(4):
            t.forward(30)
            t.left(90)
        t.end_fill()
turtle.done()
```

## ۱۱. ترسیم یک منظره ساده (خورشید و کوه)

```
import turtle
t = turtle.Turtle()
t.speed(5)
# Sun
t.penup(); t.goto(-150, 100); t.pendown()
t.color("orange", "yellow"); t.begin_fill(); t.circle(40); t.end_fill()
# Mountains
t.penup(); t.goto(-300, -100); t.pendown()
t.color("brown"); t.begin_fill()
for _ in range(3): t.forward(200); t.left(120)
t.end_fill()
turtle.done()
```

## ۱۲. الگوی اسپیروگراف (Spirograph)

```
import turtle
t = turtle.Turtle()
t.speed(0)
for i in range(100):
    t.circle(i*1.5)
    t.left(10)
turtle.done()
```

## ج) فراکتال‌ها و بازگشت (پروژه‌های ۱۳ تا ۱۷)

### ۱۳. درخت فراکتال

```
import turtle
t = turtle.Turtle()
t.left(90); t.speed(0)
def tree(d):
    if d < 10: return
    t.forward(d)
    t.left(30); tree(d*0.7); t.right(60); tree(d*0.7); t.left(30)
    t.backward(d)
tree(100)
turtle.done()
```

### ۱۴. دانه برف کخ (Koch Snowflake)

```
import turtle
t = turtle.Turtle()
t.speed(0)
def koch(d, l):
    if l == 0: t.forward(d); return
    for a in [60, -120, 60, 0]: koch(d/3, l-1); t.left(a)
for _ in range(3): koch(300, 3); t.right(120)
turtle.done()
```

## ١٥. مثلث سيربينسكي (Sierpinski Triangle)

```
import turtle
t = turtle.Turtle()
t.speed(0)
def sierpinski(l, d):
    if d == 0:
        for _ in range(3): t.forward(l); t.left(120)
    else:
        sierpinski(l/2, d-1); t.forward(l/2)
        sierpinski(l/2, d-1); t.backward(l/2); t.left(60); t.forward(l/2);
t.right(60)
        sierpinski(l/2, d-1); t.left(60); t.backward(l/2); t.right(60)
sierpinski(200, 4)
turtle.done()
```

## ١٦. منحنى هيلفرد (Hilbert Curve)

```
import turtle
t = turtle.Turtle()
t.speed(0)
def hilbert(d, a, l):
    if d == 0: return
    t.left(a); hilbert(d-1, -a, l); t.forward(l)
    t.right(a); hilbert(d-1, a, l); t.forward(l)
    hilbert(d-1, a, l); t.right(a); t.forward(l)
    hilbert(d-1, -a, l); t.left(a)
hilbert(4, 90, 10)
turtle.done()
```

## IV. حرکت تصادفی (Random Walk)

```
import turtle
import random
t = turtle.Turtle()
t.speed(0); t.width(3)
colors = ["red", "blue", "green", "purple", "orange"]
for _ in range(200):
    t.color(random.choice(colors))
    t.forward(30)
    t.setheading(random.choice([0, 90, 180, 270]))
turtle.done()
```

## د) پروژه‌های تعاملی و بازی (پروژه‌های ۱۸ تا ۲۳)

### ۱۸. مسابقه لاک‌پشت‌ها

```
import turtle
import random
screen = turtle.Screen()
colors = ["red", "blue", "green", "orange"]
turtles = []
for i in range(4):
    new_t = turtle.Turtle(shape="turtle")
    new_t.color(colors[i]); new_t.penup()
    new_t.goto(-200, 50 - i*40); turtles.append(new_t)
while True:
    for t in turtles:
        t.forward(random.randint(1, 10))
        if t.xcor() > 200: print(f"{t.pencolor()} won!"); turtle.done();
exit()
```

## ۱۹. ساده بازی پونگ (Pong)

```
import turtle
# Simplified Pong Logic
s = turtle.Screen(); s.setup(600, 400)
p = turtle.Turtle(); p.shape("square"); p.shapesize(5, 1); p.penup();
p.goto(-250, 0)
b = turtle.Turtle(); b.shape("circle"); b.penup()
dx, dy = 2, 2
while True:
    b.goto(b.xcor()+dx, b.ycor()+dy)
    if b.ycor() > 190 or b.ycor() < -190: dy *= -1
    if b.xcor() > 290 or b.xcor() < -290: dx *= -1
    if b.distance(p) < 50 and b.xcor() < -240: dx *= -1
```

## ۲۰. ساعت دیجیتال

```
import turtle
import time
t = turtle.Turtle()
t.hideturtle()
while True:
    t.clear()
    t.write(time.strftime("%H:%M:%S"), font=("Arial", 40, "normal"),
align="center")
    time.sleep(1)
```

## ۲۱. برنامه نقاشی تعاملی

```
import turtle
t = turtle.Turtle()
def move_f(): t.forward(10)
def turn_l(): t.left(10)
def turn_r(): t.right(10)
s = turtle.Screen()
s.onkey(move_f, "Up"); s.onkey(turn_l, "Left"); s.onkey(turn_r, "Right")
s.listen(); turtle.done()
```

## ساده (Snake) بازی مار ۲۲

```
import turtle
import time
import random

s = turtle.Screen(); s.setup(600, 600); s.tracer(0)
head = turtle.Turtle(); head.shape("square"); head.penup()
food = turtle.Turtle(); food.shape("circle"); food.color("red");
food.penup(); food.goto(0, 100)
segments = []
def move():
    if head.direction == "up": head.sety(head.ycor() + 20)
    # ... more logic ...
head.direction = "stop"
while True:
    s.update(); time.sleep(0.1)
```

## نتیجه‌گیری

فراتر از یک ابزار آموزشی صرف است؛ این کتابخانه دروازه‌ای به سوی درک عمیق‌تر Turtle کتابخانه مفاهیم برنامه‌نویسی و گرافیک کامپیوتری است. با تمرین بر روی این ۲۲ پروژه، شما نه تنها مهارت‌های پایتون خود را تقویت می‌کنید، بلکه توانایی خلق تصاویر و برنامه‌های بصری جذاب را نیز به دست خواهید آورد. این پروژه‌ها تنها نقطه شروع هستند و با خلاقیت شما، می‌توانند به آثار هنری و برنامه‌های پیچیده‌تری تبدیل شوند.