

intro to pygame

How use?

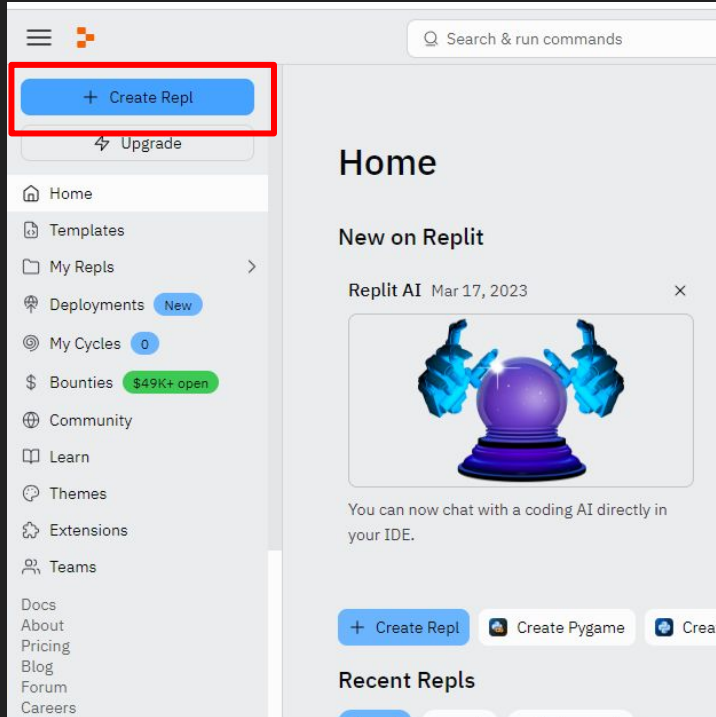
what pygame can do

- Acts as a game engine
- Automate the boring stuff
 - Create your game's window
 - Graphics
 - Tickrate

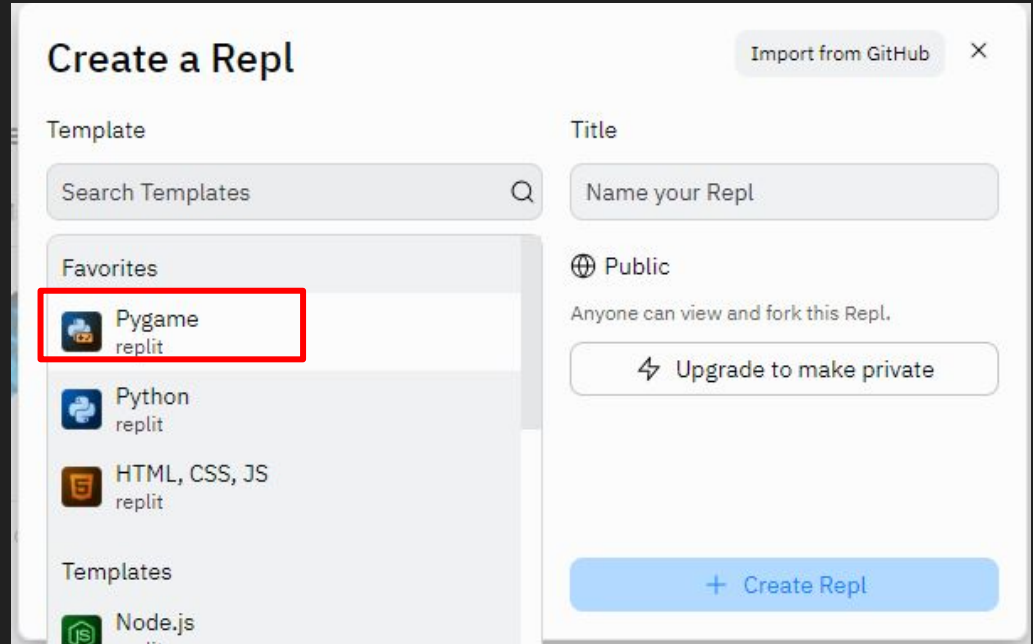
let's get to coding

create your project in repl.it

Create a new python project in repl.it.



The screenshot shows the Replit home page. A red box highlights the '+ Create Repl' button in the top navigation bar. Below it is an 'Upgrade' button. The left sidebar contains navigation links: Home, Templates, My Repls, Deployments (with a 'New' badge), My Cycles (with a '0' badge), Bounties (\$49K+ open), Community, Learn, Themes, Extensions, and Teams. The main content area features a 'Home' heading, a 'New on Replit' section with a 'Replit AI' card dated Mar 17, 2023, and a 'Recent Repls' section. At the bottom of the main content area, there are buttons for '+ Create Repl', 'Create Pygame', and 'Create Node.js'.

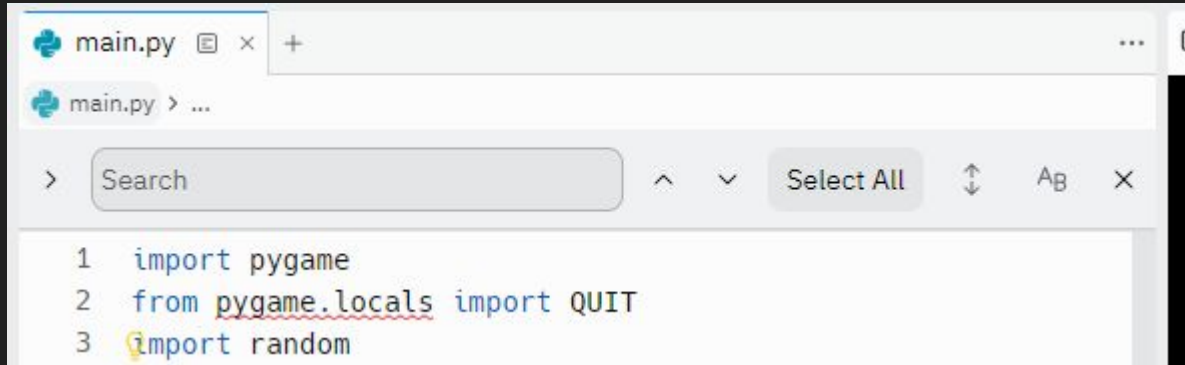


The screenshot shows the 'Create a Repl' dialog box. At the top right, there is an 'Import from GitHub' button with a close icon. The dialog is divided into two main sections: 'Template' and 'Title'. The 'Template' section has a search bar labeled 'Search Templates' and a list of templates. A red box highlights the 'Pygame replit' template in the 'Favorites' section. Other templates listed include 'Python replit' and 'HTML, CSS, JS replit'. The 'Title' section has a text input field labeled 'Name your Repl'. Below the input field, there is a 'Public' visibility selector and a description: 'Anyone can view and fork this Repl.'. At the bottom right, there is an 'Upgrade to make private' button and a '+ Create Repl' button.

installation

Type `import pygame` and run it.

Pygame should install automatically for you. (On other environments, you may need to install it using pip.)



```
main.py x +
main.py > ...
> Search ^ v Select All ⬆ ⬆ AB X
1 import pygame
2 from pygame.locals import QUIT
3 import random
```

pygame initialization

```
import pygame
```

```
pygame.init()
```

```
pygame.display.set_caption("Some Title Goes Here")
```

```
canvas = pygame.display.set_mode((1200, 800))
```

```
def main():
```

```
    pass # We'll put future code here...
```

```
main()
```

pygame initialization

Add a main loop to the main() function:

```
def main():  
    while True:  
        for event in pygame.event.get():  
            if event.type == pygame.QUIT:  
                pygame.quit()  
                exit()  
        pygame.display.update()
```

done starting!

Run it!

But it's just a black screen for now...

more features ... ?

Let's get into it!

0001. printing text on screen

Add inside the main function but before your loop:

```
def main():  
    my_font = pygame.font.SysFont('Calibri', 20)  
    rendered_text = my_font.render('This is some text', True, (0,  
0, 255))  
    canvas.blit(rendered_text, (200, 200))  
  
    while True:
```

This draws "This is some text" in blue (0, 0, 255) with the top-left corner of the box at (200, 200).

0010. drawing rectangles

Add this after the previous code but before the loop:

```
    canvas.blit(rendered_text, (200, 200))

    pygame.draw.rect(canvas, (255, 255, 255), (300, 300, 50,
50))

    while True:
```

This draws a solid white rectangle with top-left corner at 300, 300 with a *size* of 50 by 50.

0011. drawing arbitrary shapes

```
pygame.draw.rect(canvas, (255, 255, 255), (300, 300, 50, 50))
```

```
points = [(400, 400), (400, 500), (500, 500), (500, 400), (450,  
330), (400, 400)]
```

```
pygame.draw.aalines(canvas, (0, 255, 0), True, points)
```

```
while True:
```

0011. drawing arbitrary shapes

- `aa_lines` stands for antialiased lines (basically drawing it prettier)
- If drawing a closed shape, put the first point a second time at the end
- Can also use `pygame.draw.lines(..., width=...)` to draw lines with a thicker width, but they might not look as good since there is no antialiasing

0100. detecting mouse clicks

```
while True:
```

```
    for event in pygame.event.get():
```

```
        if event.type == pygame.QUIT:
```

```
            pygame.quit()
```

```
            exit()
```

```
        if event.type == pygame.MOUSEBUTTONDOWN:
```

```
            canvas.fill((0, 0, 0))
```

```
            rendered_text = my_font.render('You clicked at
```

```
' + str(event.pos), True, (0, 255, 0))
```

```
            canvas.blit(rendered_text, (300, 300))
```

```
        pygame.display.update()
```

0100. detecting mouse clicks

- Check for new events (using `pygame.event.get`)
- Mouse button press is an event (`pygame.MOUSEBUTTONDOWN`)
 - To check left/right click, can use `event.button` (left is 1, middle is 2, right is 3)
 - To get the position of the click, can use `event.pos`
- Then we clear the canvas - `canvas.fill((0, 0, 0))` fills it all black
- And finally redraw our text

0101. time

First, let's move our old code into the loop so it doesn't get deleted every click:

```
def main():
```

```
    my_font = pygame.font.SysFont('Calibri', 20)
```

```
    Cut out the code right here
```

```
    while True:
```

```
        for event in pygame.event.get():
```

```
            ...
```

```
            And put it right here after the for loop for the events
```

```
            rendered_text = my_font.render('This is some text', True, (0, 0, 255))
```

```
            canvas.blit(rendered_text, (200, 200))
```

```
            pygame.draw.rect(canvas, (255, 255, 255), (300, 300, 50, 50))
```

```
            points = [(400, 400), (400, 500), (500, 500), (500, 400), (450, 330), (400, 400)]
```

```
            pygame.draw.aalines(canvas, (0, 255, 0), True, points)
```

```
            pygame.display.update()
```


0101. time

Now add these 2 lines so our game runs at 30 FPS for now:

```
def main():  
    my_font = pygame.font.SysFont('Calibri', 20)  
    clock = pygame.time.Clock()  
  
    while True:  
        clock.tick(30)  
        for event in pygame.event.get():
```

and that's it!

These are basically all the most important parts of
pygame!

(You can make games like HexaMine)

questions?

Get coding!