

Рисование фигур

Модуль *tkinter*



Холст *Canvas*

Рисовать фигуры и размещать графические объекты можно на объекте холста *Canvas*.

При создании объекта-экземпляра *Canvas* необходимо указать его ширину, высоту и цвет.

```
c = Canvas(root, width=200, height=200, bg='#ffffff')
```

Здесь *root* – экземпляр класса *Tk*

Метод *pack* позволяет отобразить объект в окне.

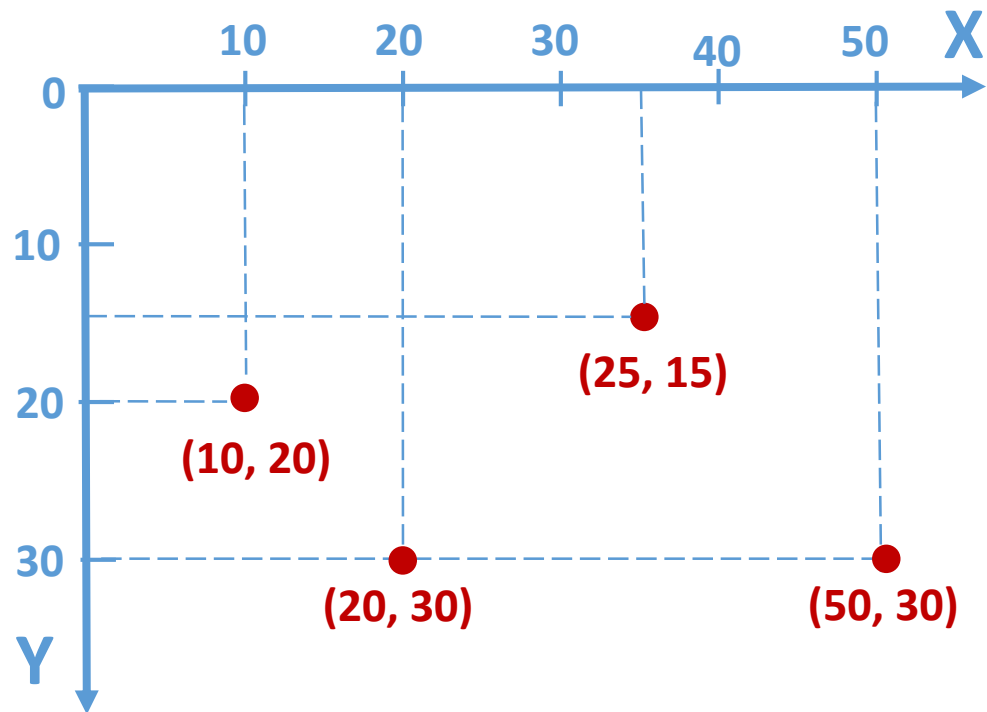
```
c.pack()
```



Система координат холста

При размещении геометрических примитивов и других объектов указываются их координаты на холсте.

Точкой отсчета является верхний левый угол.

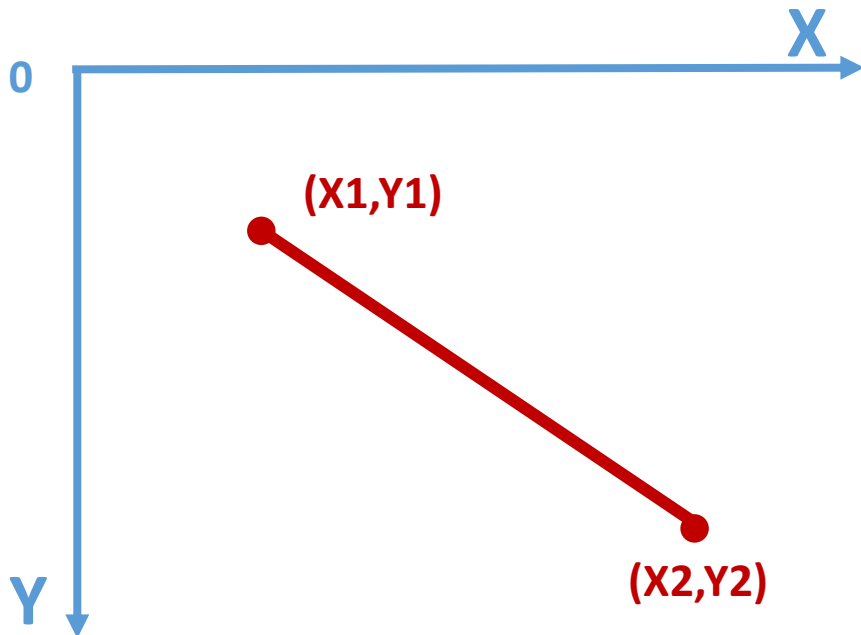




Рисование отрезка

`холст.create_line(x1, y1, x2, y2, параметры)`

Метод `create_line` рисует отрезок от точки с координатами (x_1, y_1) до точки с координатами (x_2, y_2) . Параметры могут отсутствовать.



Пары координат могут быть взяты в квадратные или круглые скобки

`холст.create_line([x1, y1], [x2, y2])`

`холст.create_line((x1, y1), (x2, y2))`



Параметры отрезка

Ширина обводки

width = ширина обводки

Цвет обводки

fill = цвет заливки

Вид пунктирной обводки

dash = (число1, число2)

число1 — длина штриха в *px*, **число2** — длина промежутка в *px*

Но!!! Шаблон тире зависит от операционной системы

Наличие стрелок

arrow = СЛОВО

СЛОВО может принимать значения:

NONE (нет стрелки), **BOTH** (стрелки на обоих концах линии),

FIRST (стрелка в начале линии), **LAST** (стрелка в конце линии)

Начало линии — первая пара координат, конец — последняя пара



```
c.create_line(50, 50, 50, 400, width=4,  
fill=line_clr)
```

```
c.create_line(100, 50, 100, 400, width=10,  
fill='#800040')
```

```
c.create_line(150, 50, 150, 400, width=4,  
fill=line_clr, dash=(10,5))
```

```
c.create_line(200, 50, 200, 400, width=4,  
fill=line_clr, dash=(100,20))
```

```
c.create_line(250, 50, 250, 400, width=4,  
fill=line_clr, dash=(20,10,10))
```

```
c.create_line(300, 50, 300, 400, width=4,  
fill=line_clr, dash=(20,10,10,20))
```



```
c.create_line(350, 50, 750, 50,  
width=4, fill=line_clr, arrow=NONE)
```



```
c.create_line(350, 120, 750, 120,  
width=4, fill=line_clr, arrow=FIRST)
```



```
c.create_line(750, 190, 350, 190,  
width=4, fill='#800040', arrow=FIRST)
```



```
c.create_line(350, 260, 750, 260,  
width=4, fill=line_clr, arrow=LAST)
```



```
c.create_line(750, 330, 350, 330,  
width=4, fill='#800040', arrow=LAST)
```



```
c.create_line(350, 400, 750, 400,  
width=4, fill=line_clr, arrow=BOTH)
```



Параметры замкнутых фигур

Ширина обводки

`width = ширина обводки`

Цвет обводки

`outline = цвет обводки`

Цвет заливки

`fill = цвет заливки`

Вид пунктирной обводки

`dash = (число1, число2)`

`число1` — длина штриха в *px*, `число2` — длина промежутка в *px*

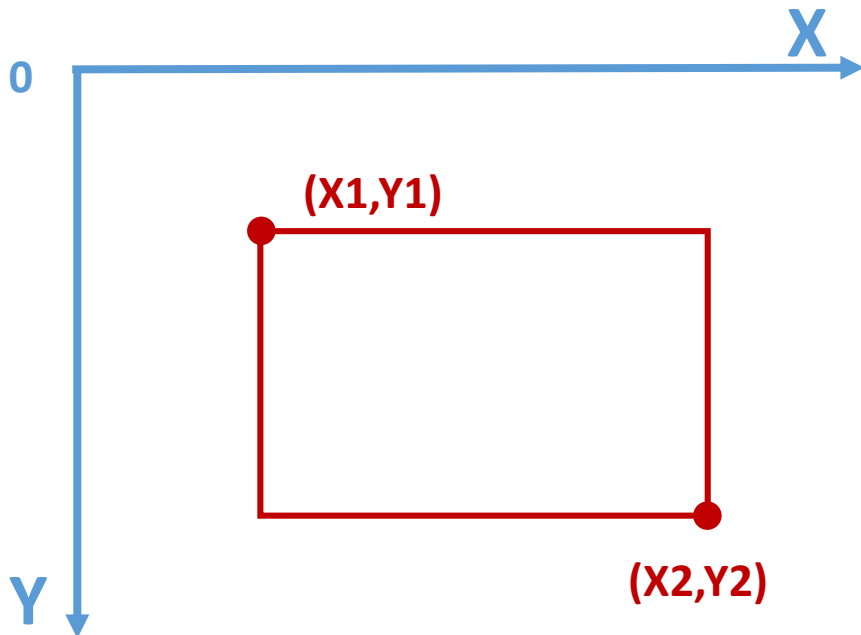
Но!!! Шаблон тире зависит от операционной системы



Рисование прямоугольника

`холст.create_rectangle(x1, y1, x2, y2, параметры)`

Метод `create_rectangle` рисует прямоугольник с координатами верхнего левого угла $(x1, y1)$ и координатами правого нижнего угла $(x2, y2)$. Параметры могут отсутствовать.



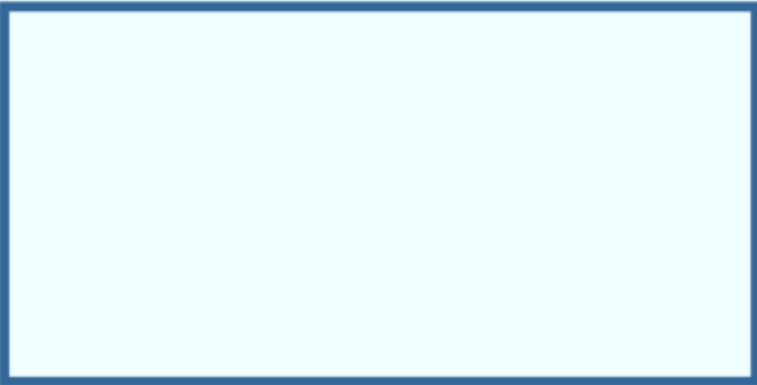
Пары координат могут быть взяты в квадратные или круглые скобки

`холст.create_rectangle([x1, y1], [x2, y2])`

`холст.create_rectangle((x1, y1), (x2, y2))`



```
c.create_rectangle(50, 50, 350, 200)
```



```
c.create_rectangle(50, 250, 350, 400,  
width=4, outline=line_clr)
```



```
c.create_rectangle([400, 50],  
[750, 200], width=4,  
outline=line_clr,  
fill='#FfCf84')
```

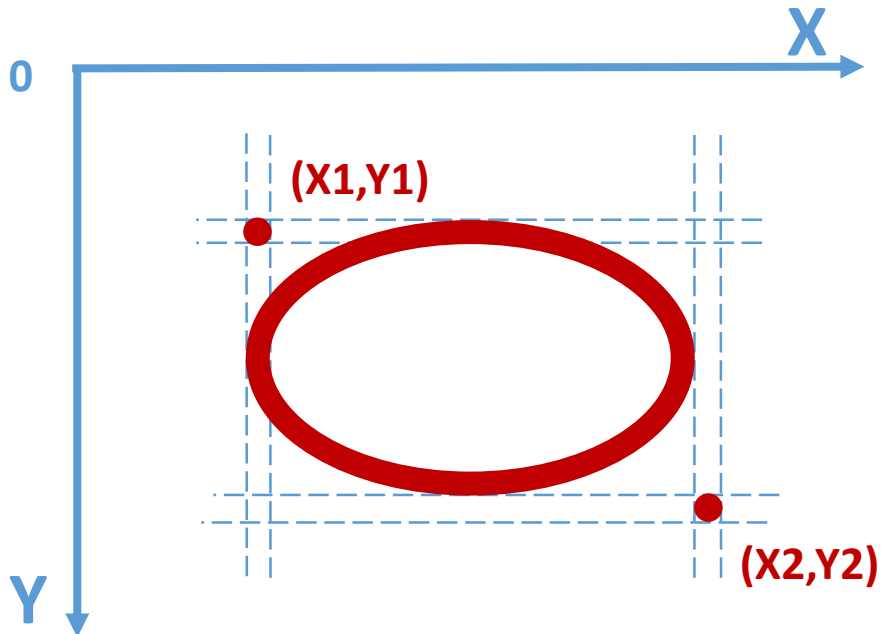
```
c.create_rectangle((400, 250),  
(750, 400), width=20,  
outline='#800040',  
fill='#FfCf84', dash=(100, 20))
```



Рисование овала

`холст.create_oval(x1, y1, x2, y2, параметры)`

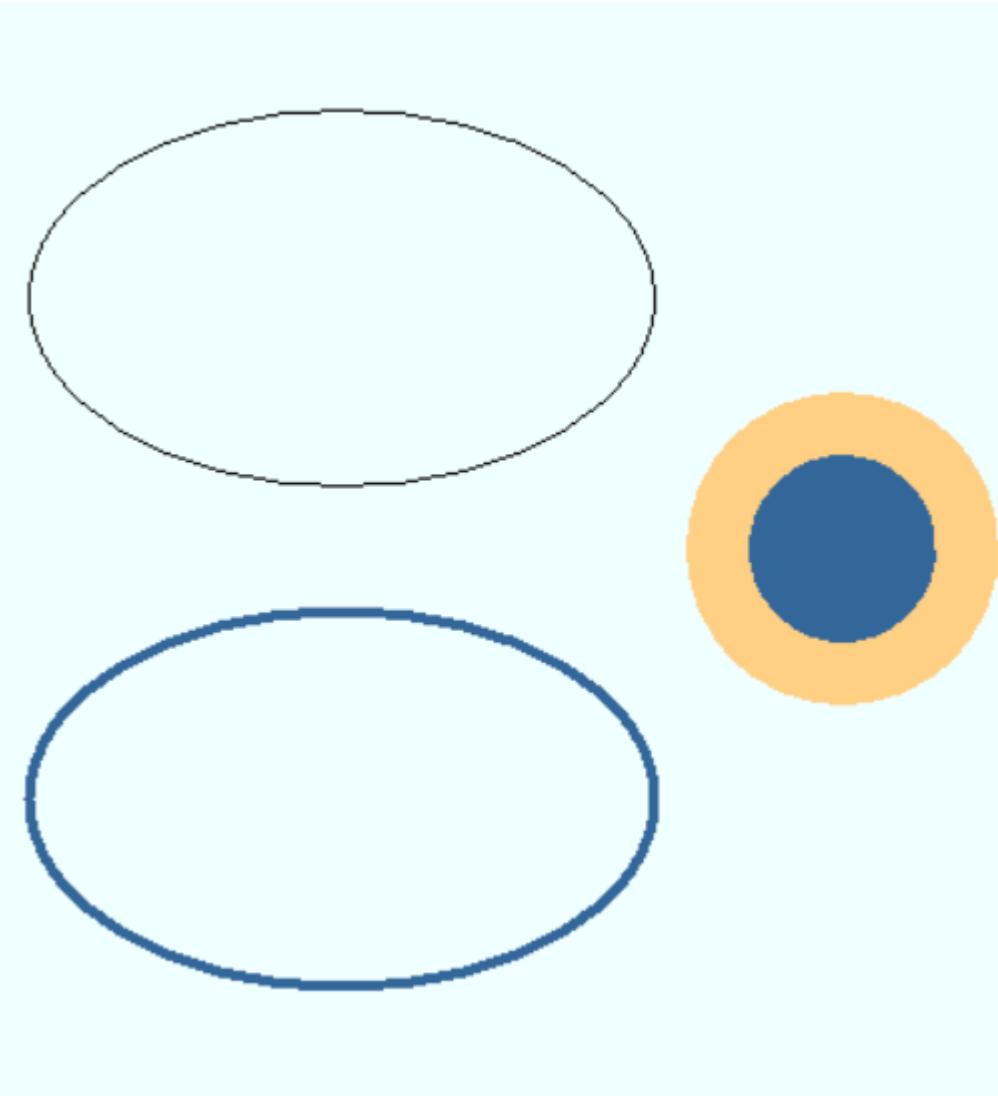
Метод `create_oval` рисует овал, вписанный в прямоугольник с координатами верхнего левого угла $(x1, y1)$ и координатами правого нижнего угла $(x2, y2)$. Параметры могут отсутствовать.



Пары координат могут быть взяты в квадратные или круглые скобки

`холст.create_oval([x1, y1], [x2, y2])`

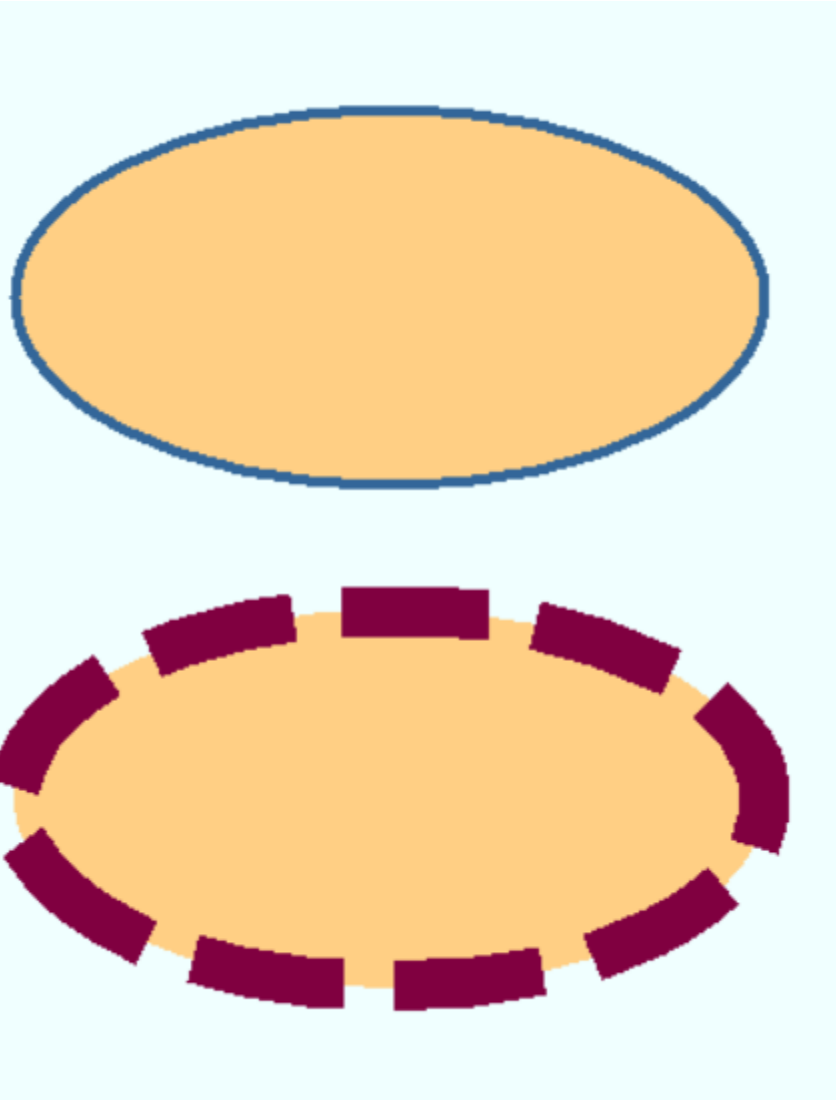
`холст.create_oval((x1, y1), (x2, y2))`



```
c.create_oval(50, 50, 300, 200)
```

```
c.create_oval(325, 175, 425, 275,  
width=25, outline='#FfCf84',  
fill=line_clr)
```

```
c.create_oval(50, 250, 300, 400,  
width=4, outline=line_clr)
```



```
c.create_oval([450,50],  
[750,200], width=4,  
outline=line_clr,  
fill='#FfCf84')
```

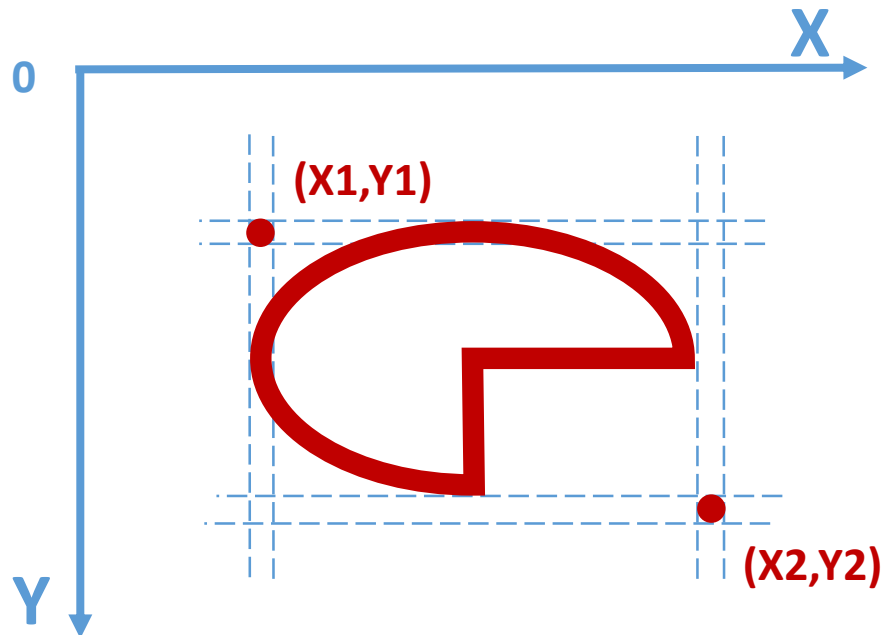
```
c.create_oval((450,250),  
(750,400), width=20,  
outline='#800040',  
fill='#FfCf84', dash=(100,20))
```



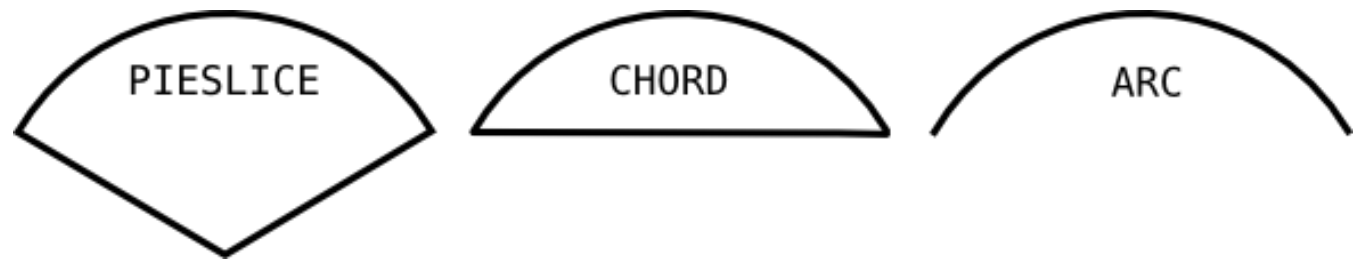
Рисование дуги

`холст.create_arc(x1, y1, x2, y2, параметры)`

Метод `create_arc` рисует дуговой объект, вырезанный из эллипса. Эллипс вписан в прямоугольник с координатами верхнего левого угла $(x1, y1)$ и координатами правого нижнего угла $(x2, y2)$. Параметры могут отсутствовать.



Параметр `style` определяет вид объекта



`style=PIESLICE` — можно не указывать, это значение по умолчанию

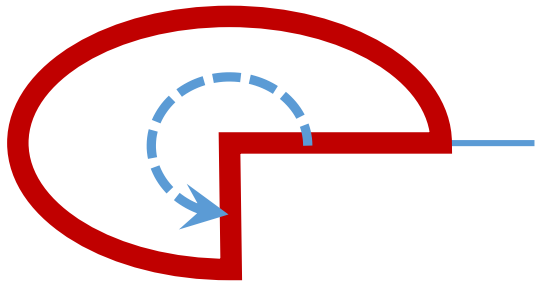


Параметры дуги

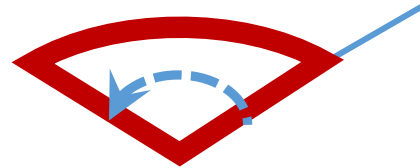
Кроме общих параметров и параметра *style* дуга имеет еще два параметра: *start* и *extent*. По умолчанию *start=0* и *extent=90*.

Положительные числа – против часовой стрелки, отрицательные – против.

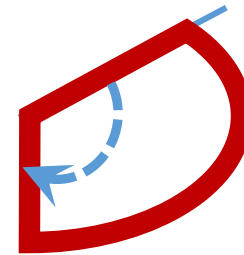
start=0, extent=270



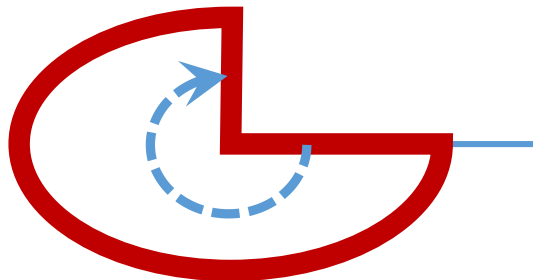
start=30, extent=120



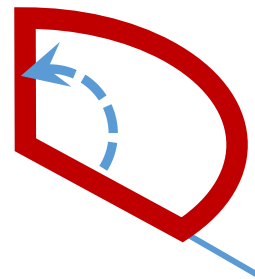
start=30, extent=-120



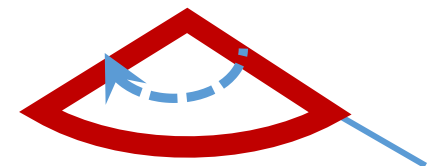
start=0, extent=-270

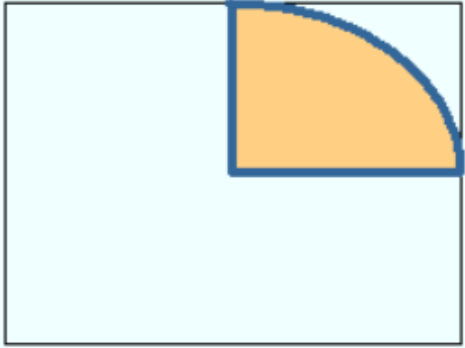


start=-30, extent=120



start=-30, extent=-120

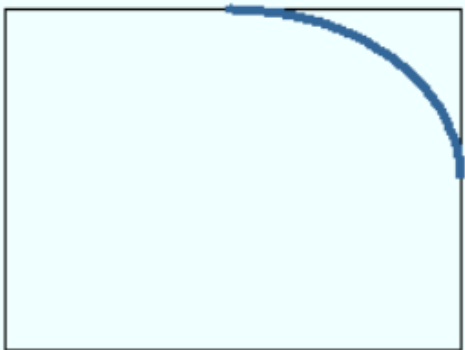




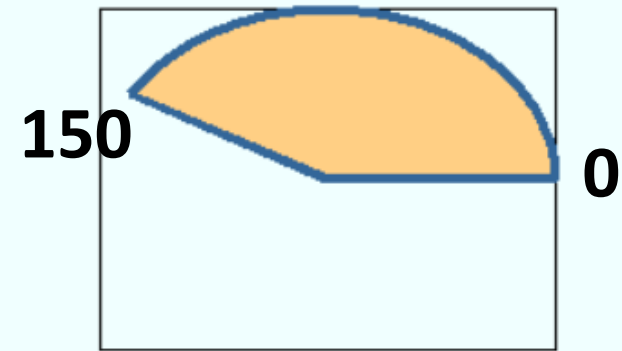
```
p1=(50,50)
p2=(250,200)
c.create_rectangle(p1, p2)
c.create_arc(p1, p2, width=4,
outline=line_clr, fill='#FfCf84')
```



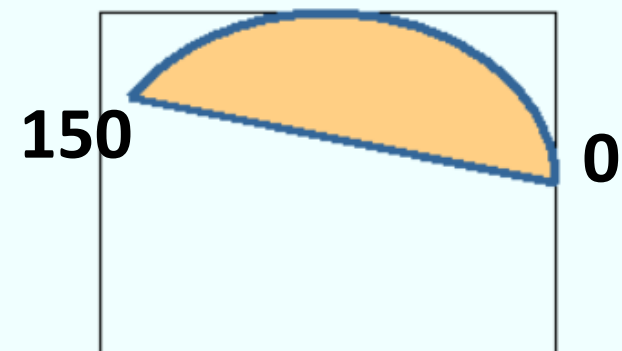
```
p1=(50,250)
p2=(250,400)
c.create_rectangle(p1, p2)
c.create_arc(p1, p2, width=4,
outline=line_clr, fill='#FfCf84', style=CHORD)
```



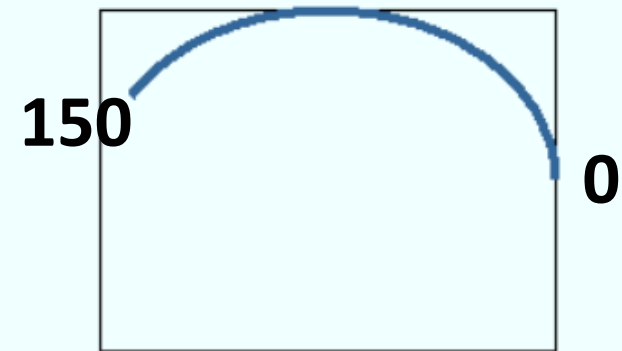
```
p1=(50,450)
p2=(250,600)
c.create_rectangle(p1, p2)
c.create_arc(p1, p2, width=4,
outline=line_clr, fill='#FfCf84', style=ARC)
```



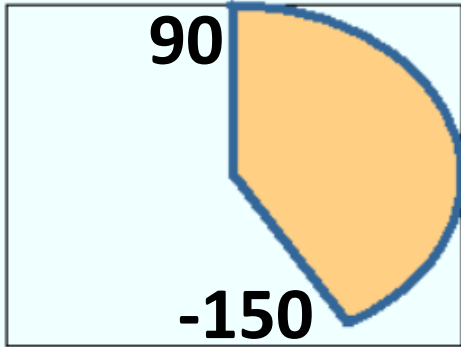
```
p1=(300,50)
p2=(500,200)
c.create_rectangle(p1, p2)
c.create_arc(p1, p2, width=4, outline=line_clr,
fill='#FfCf84', start=0, extent=150)
```



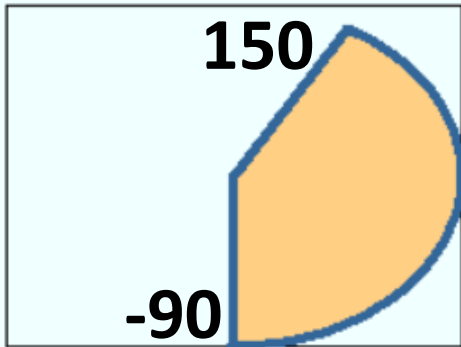
```
p1=(300,250)
p2=(500,400)
c.create_rectangle(p1, p2)
c.create_arc(p1, p2, width=4, outline=line_clr,
fill='#FfCf84', style=CHORD, start=0, extent=150)
```



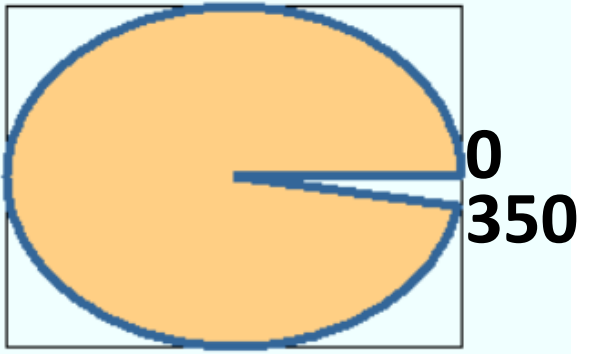
```
p1=(300,450)
p2=(500,600)
c.create_rectangle(p1, p2)
c.create_arc(p1, p2, width=4, outline=line_clr,
fill='#FfCf84', style=ARC, start=0, extent=150)
```



```
p1=(550,50)
p2=(750,200)
c.create_rectangle(p1, p2)
c.create_arc(p1, p2, width=4, outline=line_clr,
fill='#FfCf84', start=90, extent=-150)
```



```
p1=(550,250)
p2=(750,400)
c.create_rectangle(p1, p2)
c.create_arc(p1, p2, width=4, outline=line_clr,
fill='#FfCf84', start=-90, extent=150)
```



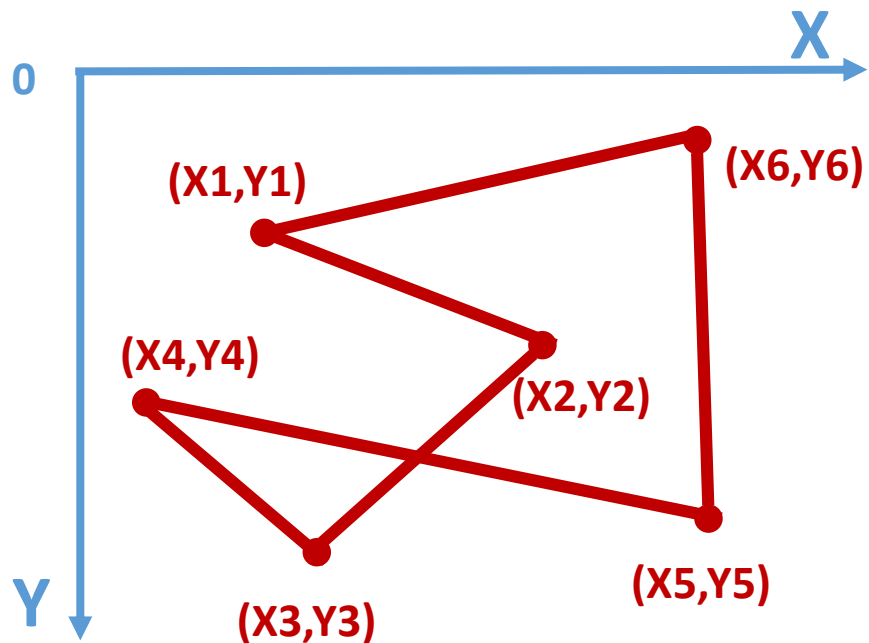
```
p1=(550,450)
p2=(750,600)
c.create_rectangle(p1, p2)
c.create_arc(p1, p2, width=4, outline=line_clr,
fill='#FfCf84', start=0, extent=350)
```



Рисование многоугольника

`холст.create_polygon(x1, y1, x2, y2, ..., xN, yN, параметры)`

Метод `create_polygon` рисует многоугольник по координатам. Координатных точек должно быть как минимум три. Параметры могут отсутствовать.



Пары координат могут быть взяты в квадратные или круглые скобки

`холст.create_polygon([x1, y1], ..., [xN, yN])`

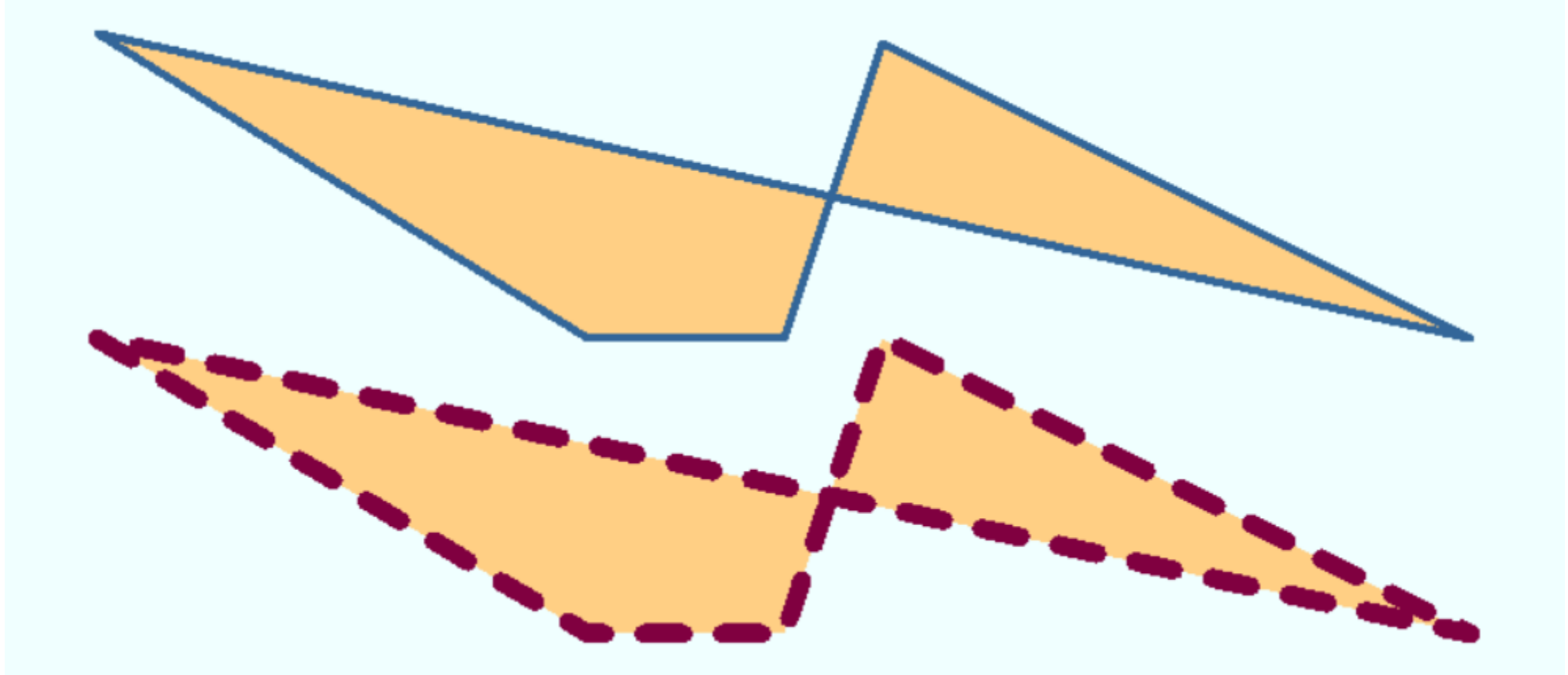
`холст.create_polygon((x1, y1), ..., (xN, yN))`



```
c.create_polygon(15, 50, 215,25, 315, 100,  
115, 100, outline='#ff0000')
```

```
c.create_polygon(275, 50, 475,25, 575, 100,  
375, 100, fill=line_clr)
```

```
c.create_polygon(550, 50, 725,25, 790, 100,  
650, 100, fill='', width=6, outline=line_clr)
```



```
c.create_polygon((50,120), (300,275), (400,275), (450,125),  
(750,275), width=4, outline=line_clr, fill='#FfCf84')
```

```
c.create_polygon([50,275], [300,425],[400,425], [450,275],  
[750,425], width=10, outline='#800040', fill='#FfCf84',  
dash=(100,20))
```



Все рассмотренные примеры рисования фигур находятся в файлах:

- *PA_lines.py*
- *PA_rect.py*
- *PA_oval.py*
- *PA_arc.py*
- *PA_poly.py*