```python
import RPi.GPIO as GPIO # GPIOを利用する
import time             # sleepを利用する

# ポート番号の定義
SWITCH = 18
LED = 21
led_value = GPIO.LOW

# GPIOの初期化 --- (*1)
GPIO.setmode(GPIO.BCM)
GPIO.setup(SWITCH, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(LED, GPIO.OUT)

# スイッチの切り替えイベントの定義 --- (*2)
def callback_change_switch(ch):
    global led_value
    print("callback", ch)
    if ch != SWITCH: return
    #違うボタンが押されたときに何もしないためのif文
    if led_value == GPIO.LOW:
        GPIO.output(LED, )#①この行のプログラムを完成させる
        led_value = GPIO.HIGH
    else:
        GPIO.output(LED, )#②この行のプログラムを完成させる
        led_value = GPIO.LOW

# イベントの設定 --- (*3)
GPIO.add_event_detect(
    SWITCH, # ポート番号
    GPIO.RISING, # イベントの種類
    callback=callback_change_switch, # 関数の指定
    bouncetime=200) # 連続イベントを制限

# LEDを消灯する
GPIO.output(LED, GPIO.LOW)

# プログラムが終わらないように待機 --- (*4)
try:
    while True:
        time.sleep(0.1)

except KeyboardInterrupt:
    GPIO.cleanup()
```

```python
import RPi.GPIO as GPIO # GPIOを利用する
import time             # sleepを利用する

# ポート番号の定義
SWITCH = 18
LED = 21
led_value = GPIO.LOW

# GPIOの初期化 --- (*1)
GPIO.setmode(GPIO.BCM)
GPIO.setup(SWITCH, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(LED, GPIO.OUT)

# スイッチの切り替えイベントの定義 --- (*2)
def callback_change_switch(ch):
    global led_value
    print("callback", ch)
    if ch != SWITCH: return
    if led_value == GPIO.LOW:
        led_value = GPIO.HIGH
    else:
        led_value = GPIO.LOW

# イベントの設定 --- (*3)
GPIO.add_event_detect(
    SWITCH, # ポート番号
    GPIO.RISING, # イベントの種類
    callback=callback_change_switch, # 関数の指定
    bouncetime=200) # 連続イベントを制限

# LEDを消灯する
GPIO.output(LED, GPIO.LOW)

# プログラムが終わらないように待機 --- (*4)
try:
    while True:
        if led_value == GPIO.HIGH:

        else:

        time.sleep(0.1)

except KeyboardInterrupt:
    GPIO.cleanup()
```

```python
import RPi.GPIO as GPIO # GPIOを利用する
import time             # sleepを利用する

# ポート番号の定義
SWITCH = 18
LED = 21
led_value = GPIO.LOW
LED_BLINKING = 1
LED_LIGHTING = 2#モードの定義

# GPIOの初期化 --- (*1)
GPIO.setmode(GPIO.BCM)
GPIO.setup(SWITCH, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(LED, GPIO.OUT)


#コールバック関数
def callback_change_switch(ch):
    global led_value
    print("callback", ch)
    if ch != SWITCH: return
    if led_mode == LED_BLINKING:
        led_mode = LED_LIGHTING
    else:
        led_mode = LED_BLINKING


# イベントの設定 --- (*3)
GPIO.add_event_detect(
    SWITCH, # ポート番号
    GPIO.RISING, # イベントの種類
    callback=callback_change_switch, # 関数の指定
    bouncetime=200) # 連続イベントを制限

# LEDを消灯する
GPIO.output(LED, GPIO.LOW)

# プログラムが終わらないように待機 --- (*4)
try:
    while True:
        if mode ==:
            GPIO.output(LED, GPIO.HIGH)
            time.sleep(0.5)
            GPIO.output(LED, GPIO.LOW)
            time.sleep(0.5)
        else:
            GPIO.output(LED, GPIO.HIGH)
            time.sleep(0.5)

except KeyboardInterrupt:
    GPIO.cleanup()

```

```python
import RPi.GPIO as GPIO # GPIOを利用する
import time             # sleepを利用する

# ポート番号の定義
SWITCH = 18
LED = 21
led_value = GPIO.LOW
LED_BLINKING = 1#モードの定義
LED_LIGHTING = 2#モードの定義

# GPIOの初期化 --- (*1)
GPIO.setmode(GPIO.BCM)
GPIO.setup(SWITCH, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(LED, GPIO.OUT)

# LEDを消灯する
GPIO.output(LED, GPIO.LOW)

# プログラムが終わらないように待機 --- (*4)
try:
    while True:
        if GPIO.input(SWITCH):#ポーリング
            mode = LED_BLINKING#点滅
        else;
            mode = LED_LIGHTING#点灯

        if mode == LED_BLINKING:
            GPIO.output(LED, GPIO.HIGH)
            time.sleep(0.5)
            GPIO.output(LED, GPIO.LOW)
            time.sleep(0.5)
        else:
            GPIO.output(LED, GPIO.HIGH)
            time.sleep(0.5)

except KeyboardInterrupt:
    GPIO.cleanup()
```