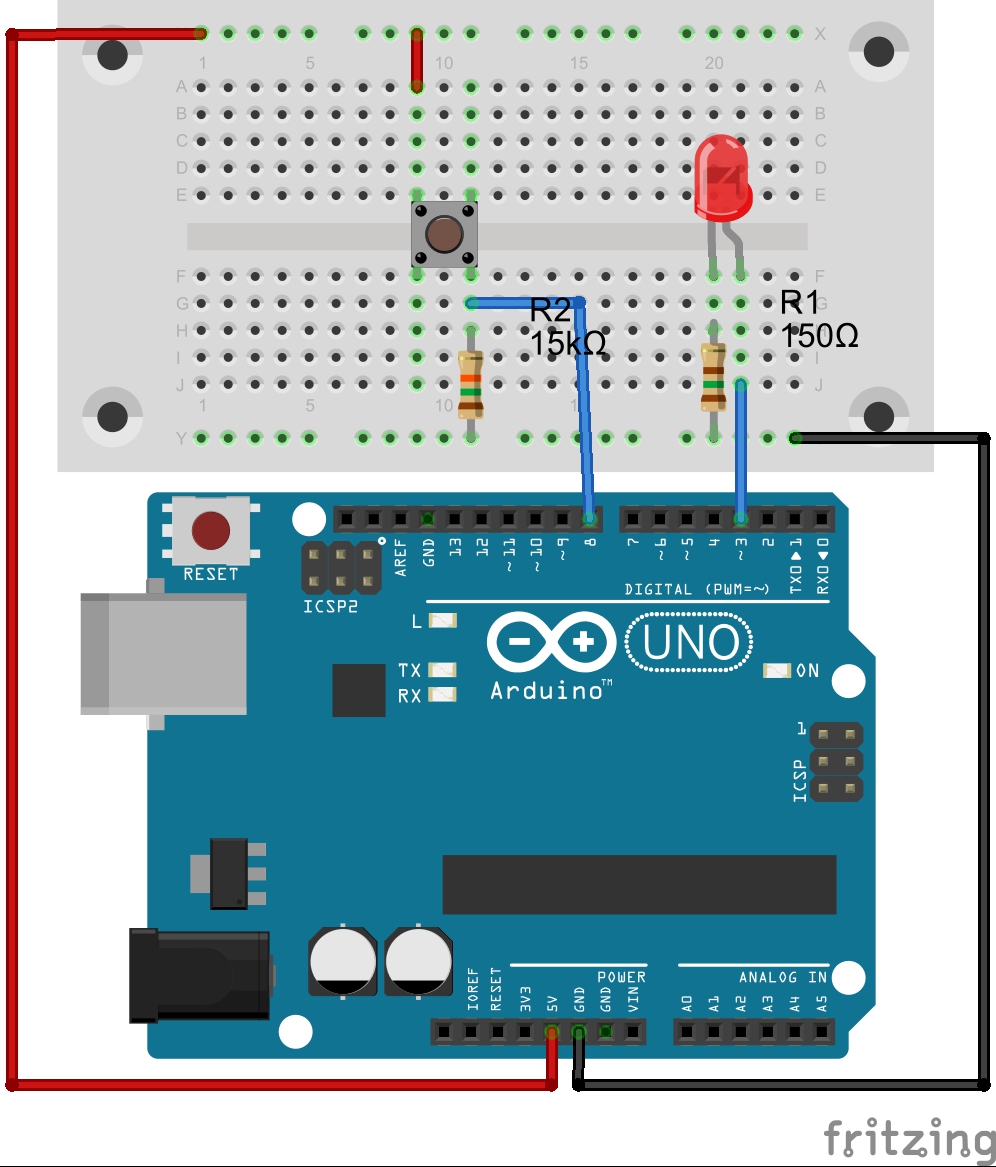
02 Digital Read

# The circuit



# The code

int LEDPin = 3; //variable storing pin number the LED is attached to

int buttonPin = 8; //variable storing pin number the button is attached to

boolean buttonState = LOW; //a variable to store the state of the button

void **setup**(){

  pinMode(LEDPin, OUTPUT);//Set the pin the LED is attached to as an output

  pinMode(buttonPin, INPUT);//Set the pin the button is attached to as an input

}//end of: setup

void **loop**(){

  buttonState = digitalRead(buttonPin); //Read the state of the button and store it in

//the buttonState variable

  if(buttonState == HIGH){ //See if the button state is hig (pressed)

    digitalWrite(LEDPin, HIGH); //Bring the LEDPin high to turn on the LED

  }

  else{ //otherwise the button is not being pressed

    digitalWrite(LEDPin, LOW); //Bring the LEDPin low to turn off the LED

  }

  delay(100); //Pause for 100 milliseconds (0.1s)

}//end of: loop

# What Next?

1. Connect a piezo buzzer to one of the digital pins and get it to play a tone when the button is pushed (have a look at ‘Help-Reference’ for tone to find out how it works. Here is the syntax you need: tone(pin, frequency, duration); )
2. Now you have one button playing one tone… try adding another button and get it to play a different tone (you can use the same piezo buzzer)
3. Maybe it would be better if you starting playing a tone without a specified duration using: tone(pin, frequency);, you could then use noTone(pin); when the button is released
4. A concept called button bounce can be an issue. Find out what it is and look at ways you can overcome it with hardware (electrical components) and software (code)