

Noms : _____
 Prénoms : _____
 Classe : _____
 Date : _____

Note : /20



Objectifs du TP

L'objectif de ce TP est de découvrir les bases de la programmation :

- séquences d'instructions,
- appels de fonctions,
- boucles répéter,
- instructions conditionnelles.

Vous programmerez des objets utilisant des boutons, LEDs, buzzer, écran, capteur de distance et servomoteurs.

Barème

Attitude, quantité de travail, qualité du compte-rendu	/1
Buzzer (Q1, Q2)	/3
Diodes (Q3, Q4, Q5)	/4
Ecran (Q6, Q7)	/3
Capteur à ultrasons (Q8)	/2
Servomoteur (Q9)	/2
Score	/5

Introduction

1. Aller sur le site web :
<https://amazon.quick-pi.org/>

- Cliquer sur le parcours 1
- Sélectionnez le langage « Blockly » :

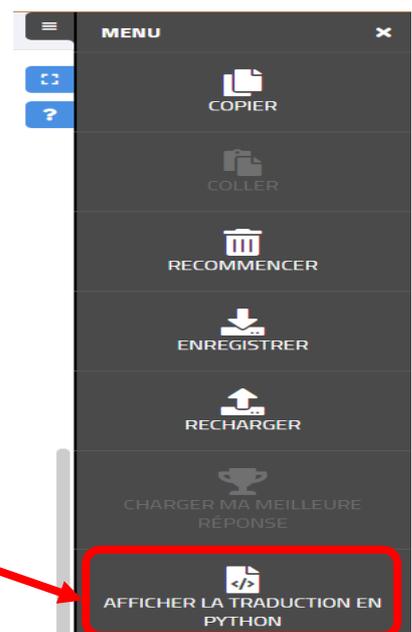
Blockly →		Ce que nous vous conseillons pour cette activité.
Scratch →		Si vous avez l'habitude de Scratch. Attention : ne fonctionne bien qu'avec les navigateurs Google Chrome ou Mozilla Firefox récents.
Python →	<pre>droite() droite() for loop in range (haut() gauche() if (obstacleBot bas())</pre>	Si vous maîtrisez bien ce langage. Attention : ne fonctionne bien qu'avec les navigateurs Google Chrome ou Mozilla Firefox récents.



- Vous devez réaliser un compte-rendu numérique qui sera à me transmettre à la fin de la séance. Votre score, qui s'affiche sur l'interface sera également pris en compte dans votre note.

Introduction ☆☆☆☆	1. Mélodie ☆☆☆☆	2. Alternance ☆☆☆☆	3. Show lumineux 1 ☆☆☆☆
4. Quelle direction ? ☆☆☆☆	5. Instrument ☆☆☆☆	6. Show lumineux 2 ☆☆☆☆	7. Avertisseur ☆☆☆☆
	8. Servo chronométré ☆☆☆☆	Donnez votre avis ☆☆☆☆	

- Pour afficher vos programmes en Python, cliquez sur « menu » puis sur « Afficher la traduction en Python ».



1. Utilisation d'un buzzer

Q1) Compléter le programme en Python qui joue la mélodie : **“mi ré mi ré do”**, en jouant chaque note pendant 500ms, l'une après l'autre, puis éteint le buzzer.

Les fréquences des notes sont :

- Do : 523Hz,
- Ré : 587Hz,
- Mi : 659Hz.

```

setBuzzerNote("buzzer1", _____);
sleep(_____);
turnBuzzerOff();
  
```

Q2) Indiquer, en mettant des croix dans le tableau, si le buzzer par rapport à la carte de programmation est :

	INPUT	OUTPUT	ANALOG	DIGITAL
Buzzer				

2. Pilotage des diodes

Q3) Compléter le programme en Python qui permet de faire clignoter 10 fois la diode à une fréquence de 1 Hz (diode allumée pendant 0,5 seconde puis éteinte pendant 0,5 seconde).

```

forloop in range(_____):
    turnLed _____();
    sleep(_____);
    turnLed _____();
    sleep(_____);
  
```

Q4) Complétez le programme qui allume en alternance les LED rouge et bleue pendant 500ms chacune : la rouge pendant 500ms, la bleue pendant 500ms, etc. Chacune des LEDs doit être allumée 5 fois au total. Tout doit être éteint à la fin.

```

forloop in range( ):
  setLedState("red1", );
  setLedState("blue1", );
  sleep( );
  setLedState("red1", );
  setLedState("blue1", );
  sleep( );
  setLedState("blue1", );
  
```

Q5) Indiquer, en mettant des croix dans le tableau, si une diode par rapport à la carte de programmation est :

	INPUT	OUTPUT	ANALOG	DIGITAL
Diode				

3. Utilisation d'un écran

Q6) Complétez le programme qui :

- affiche le texte "Direction ?" dès le début
- puis lorsque l'on appuie sur une direction, affiche le texte "Haut", "Droite", "Bas" ou "Gauche" selon la direction appuyée.

```

displayText(" ");
while :
  ifreportBlockValue('cLR_KJL;Xm!jB9#!Uw*a',
  isButtonPressed("stick1.up")):
    displayText(" ");
  ifreportBlockValue('yA3ji;onb?TX9b5+{Ps5',
  isButtonPressed("stick1.right")):
    displayText(" ");
  ifreportBlockValue('a2Kp`5O!*NR4.Q?)c6k~',
  isButtonPressed("stick1.down")):
    displayText(" ");
  ifreportBlockValue('cHi3Nlay)Bx1G(:@nID9',
  isButtonPressed("stick1.left")):
    displayText(" ");
  
```

Q7) Complétez le tableau suivant

	INPUT	OUTPUT	ANALOG	DIGITAL
Ecran				
Bouton poussoir				
Joystick (manette)				

1. Utilisation d'un capteur à ultrasons

Q8) Compléter le programme qui allume le buzzer quand le capteur détecte un objet à moins de 500cm.

Le buzzer doit jouer un son à une fréquence égale à la distance de l'objet en centimètres, si le bouton n'est pas enfoncé et il doit jouer un son à une fréquence égale au double de la distance mesurée si le bouton est enfoncé.

Il doit être éteint s'il n'y a pas d'objet à moins de 500cm.

```

while _____:
    if (reportBlockValue('M9-uTXzf(roXuQ{zzfi',
        readDistance("distance1"))) < _____:
        ifreportBlockValue('ujr_GM(=iPJWAPLX{xWM',
            isButtonPressed()):
            setBuzzerNote("buzzer1",
                ((reportBlockValue('Tm93i=.8m_jeR~ItCLhU',
                    readDistance("distance1"))) _____));
        else:
            setBuzzerNote("buzzer1",
                (reportBlockValue('9`BmY#=4W08.xs+z#=W}',
                    readDistance("distance1"))));
        else:
            turn _____();
    
```

2. Utilisation d'un servomoteur

Q9) Écrivez un programme qui met l'angle du servomoteur à 0°, puis lorsque l'on appuie sur le bouton :

- Augmente l'angle 18 fois de 10°, toutes les 50ms
- Joue un bip pendant 500ms
- Remet l'angle à 0°

```

setServoAngle("servo1", 0);
whileTrue:
    ifreportBlockValue('/HcjOZk}IUq{DwVTiG',
        isButtonPressed()):
        forloop in range(_____):
            setServoAngle("servo1",
                ((reportBlockValue('mKL0IBV3li#|EjvCfGBV',
                    getServoAngle("servo1"))) + _____));
            sleep(_____);
            turnBuzzerOn();
            sleep(_____);
            turnBuzzer_____();
        setServoAngle("servo1", 0);
    
```