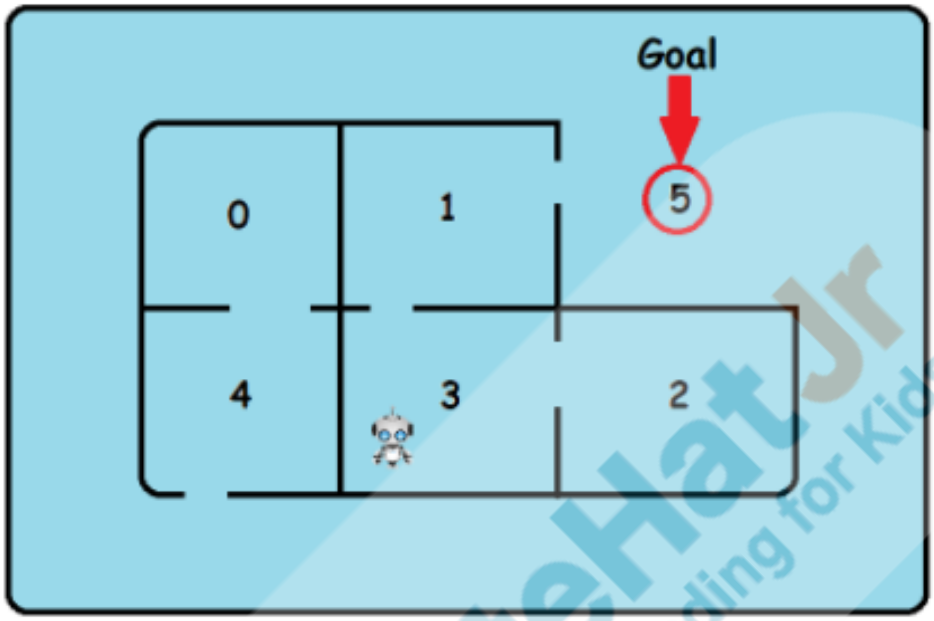
| Topic | BASIC RL COMPONENTS | |
|---|---|---|
| **Class Description** | **The student will learn the concept of Reinforcement Learning and its basic components.** | |
| **Class** | **PRO C123** | |
| **Class time** | **45 mins** | |
| **Goal** | • Introduction Reinforcement learning and its components<br>• Introduction and creation of reward matrix | |
| **Resources Required** | • Teacher Resources:<br>  ○ Laptop with internet connectivity<br>  ○ Earphones with mic<br>  ○ Notebook and pen<br>  ○ Smartphone<br>• Student Resources:<br>  ○ Laptop with internet connectivity<br>  ○ Earphones with mic<br>  ○ Notebook and pen | |
| **Class structure** | **Warm-Up**<br>**Teacher-Led Activity 1**<br>**Student-Led Activity 1**<br>**Wrap-Up** | **10 mins**<br>**10 mins**<br>**20 mins**<br>**05 mins** |
| **WARM-UP SESSION - 10 mins** | | |
| **Teacher Starts Slideshow**<br>**Slide 1 to 4**<br>Refer to speaker notes and follow the instructions on each slide. | | |
| **Teacher Action** | | **Student Action** |

| | |
|---|---|
| Hey <student's name>. How are you? It's great to see you!<br>Are you excited to learn something new today?<br><br>**Following are the WARM-UP session deliverables:**<br>• Greet the student.<br>• Revision of previous class activities.<br>• Quizzes. | **ESR**: Hi, thanks!<br>Yes, I am excited about it!<br><br>Click on the slide show tab and present the slides |
| **WARM-UP QUIZ**<br>Click on In-Class Quiz | |
| **Continue WARM-UP Session**<br>**Slide 5 to 14**  | |
| **Activity Details**<br><br>**Following are the session deliverables:**<br>• Appreciate the student.<br>• Narrate the story by using hand gestures and voice modulation methods to bring more interest in students. | |
| **Teacher Action** | **Student Action** |
| **Supervised Learning vs Reinforcement Learning:**<br><br>In the previous classes, we trained the machine learning models using the training dataset. The dataset had labels that were used to determine the category of the data. This is known as **Supervised Learning**.<br><br>**Supervised Learning** is a machine learning process that uses labeled data to learn (or get trained) and predicts output.<br><br>Today we are going to learn a new method for training a machine. In this method instead of giving a predefined dataset to the machine, it will learn by exploring the data | |

| | |
|---|---|
| collected from the environment.<br><br>Let me give you an example. There are five paths given to you. If I ask you to find the paths by which you can reach your school?<br><br>Yes!! So it'll be a trial and error method that you'll be following.<br><br>For each correct path, you will be getting a positive reward and for following the wrong path you will get a negative reward.<br><br>Let's assume that you were able to find three such paths by which you can reach your school from home. If I ask you to take anyone from it and go to school, which one will you take?<br><br>Correct!! Because you have experimented with all the paths and you will follow the one that takes minimum time.<br><br>Thus, we can conclude that:<br>1. We, humans, learn from trial and error methods.<br>2. We love positive rewards.<br>3. We make decisions depending upon our experiences.<br><br>So, instead of making a machine learn from the dataset and then give predictions or outputs if we can create a model where the machine is experimenting (trial and error). For each correct move, rewards are given. Depending upon these rewards it's making decisions.This method is known as **Reinforcement Learning(RL)**<br><br>**Reinforcement Learning** is the machine learning process machine learns by exploring from the **trial** and error **method** and takes the feedback from its own experiences.<br><br>How does this sound? | **ESR:** Follow all the paths and find the one that takes me to the school.<br><br>**ESR:** The one which takes the least amount of time to reach the school |

| | |
|---|---|
| OK. So, let's continue to implement Reinforcement learning for a machine. | **ESR:** Interesting |
| **Teacher Ends Slideshow**  | |
| **TEACHER-LED ACTIVITY - 10 mins** | |
| **Teacher Initiates Screen Share** | |
| **ACTIVITY**<br>• **To explain the basic components of the Reinforcement Learning model** | |
| **Teacher Action** | **Student Action** |
| So, we'll be looking at different components of Reinforcement learning using a similar example.<br><br>As you can see in the image, the machine or robot is present in a room lobby. The layout of a room lobby has different sections present. It has to collect different items from different sections and keep them in a bag which is marked as the Goal.<br><br>The machine has to try different paths to reach the goal and has to learn from the rewards it'll be getting. | |




The components of Reinforcement learning are

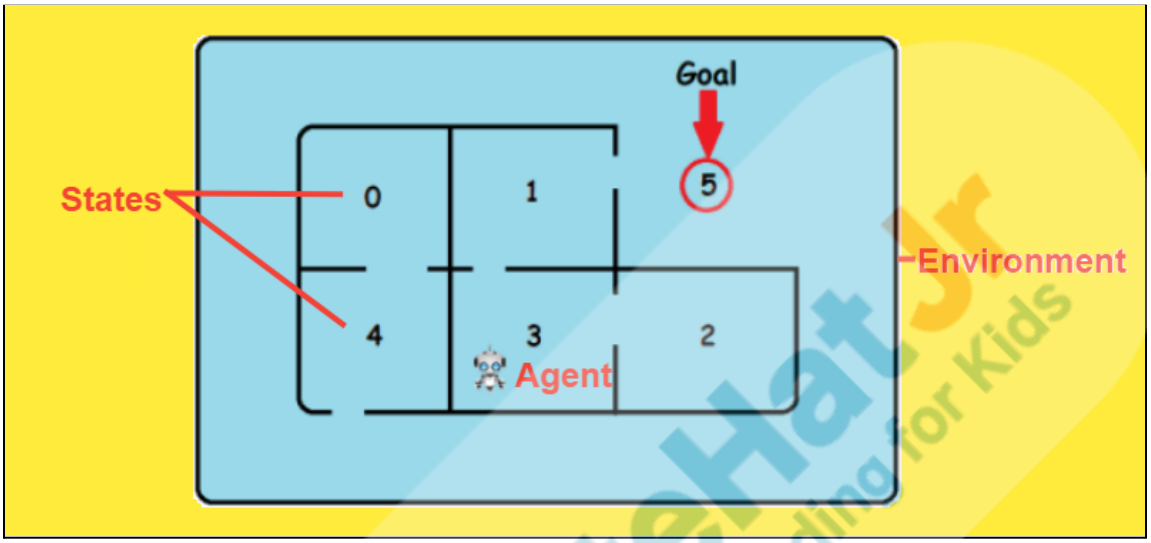
1. **Agent**
2. **Environment**
3. **States**
4. **Actions**
5. **Rewards**

Let's learn these components in the context of Reinforcement learning.

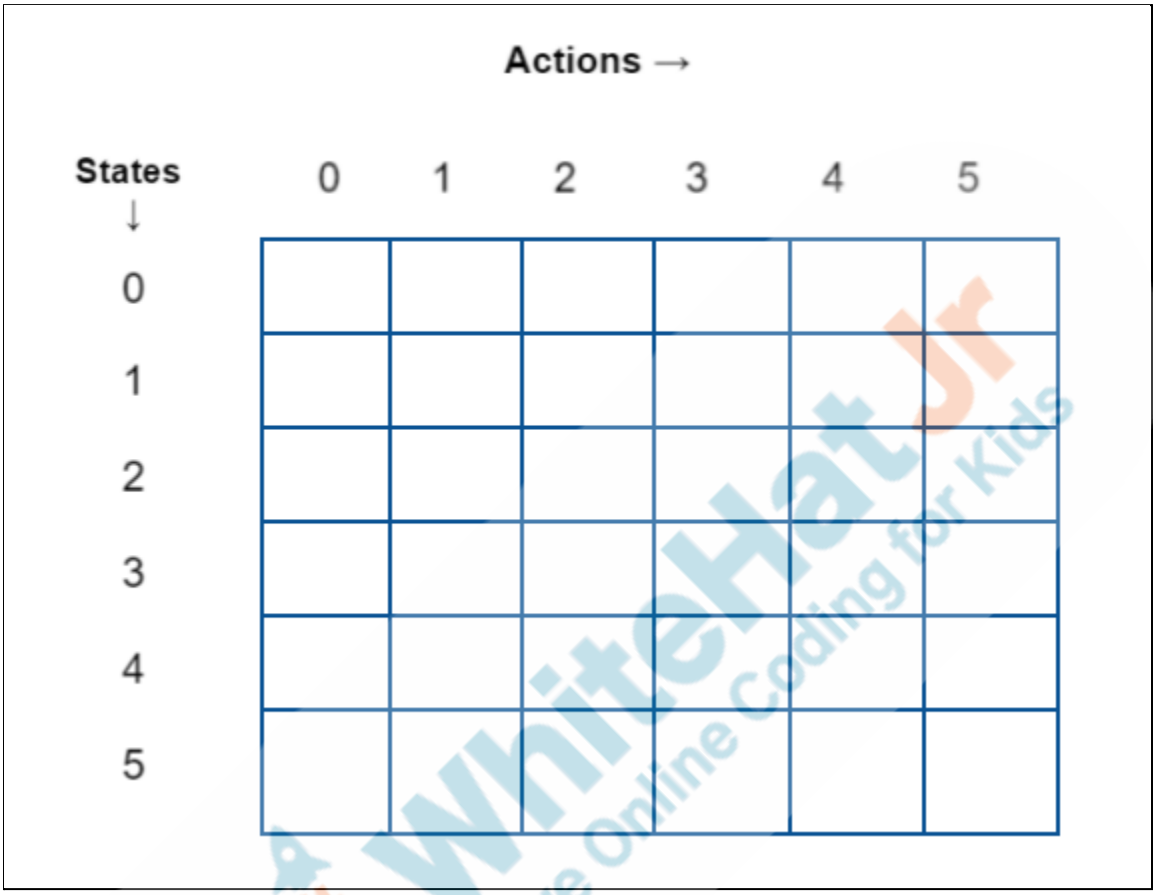
The very first component of RL is the machine. It is known as **Agent**.

Now, the Agent is interacting with the grocery shop so it is called the **Environment.**

Different sections of the grocery shops are known as **States**. The states are defined to trace the position of the agent. Here we have six sections. The numbering can be given randomly. Here we are giving numbers from 0 to 5.

| | |
|---|---|
| ***Note:*** *The numbers assigned to states are random.* | |
|  | |
| Now, to reach the Goal the agent has to move through these states. These are known as **Actions**.<br><br>For each **Action** towards the **Goal**, the agent will be assigned either **positive or negative points**. These points are known as **Rewards**. | |
| Now, the agent can be in any state. It has to take action to move through these states to reach **state 5** which is the **goal**. For each right move/action, the agent will be given a positive reward else it will get a negative reward.<br><br>The aim is to maximize the rewards and find the best path for the agent to move through these states. To calculate the reward we'll be creating a matrix which is known as the **Reward matrix.**<br><br>Reward matrix is used to give rewards to the agent and let the agent know about its performance.<br><br>Let's decide possible Rewards as **-1, 0, 100:** | |

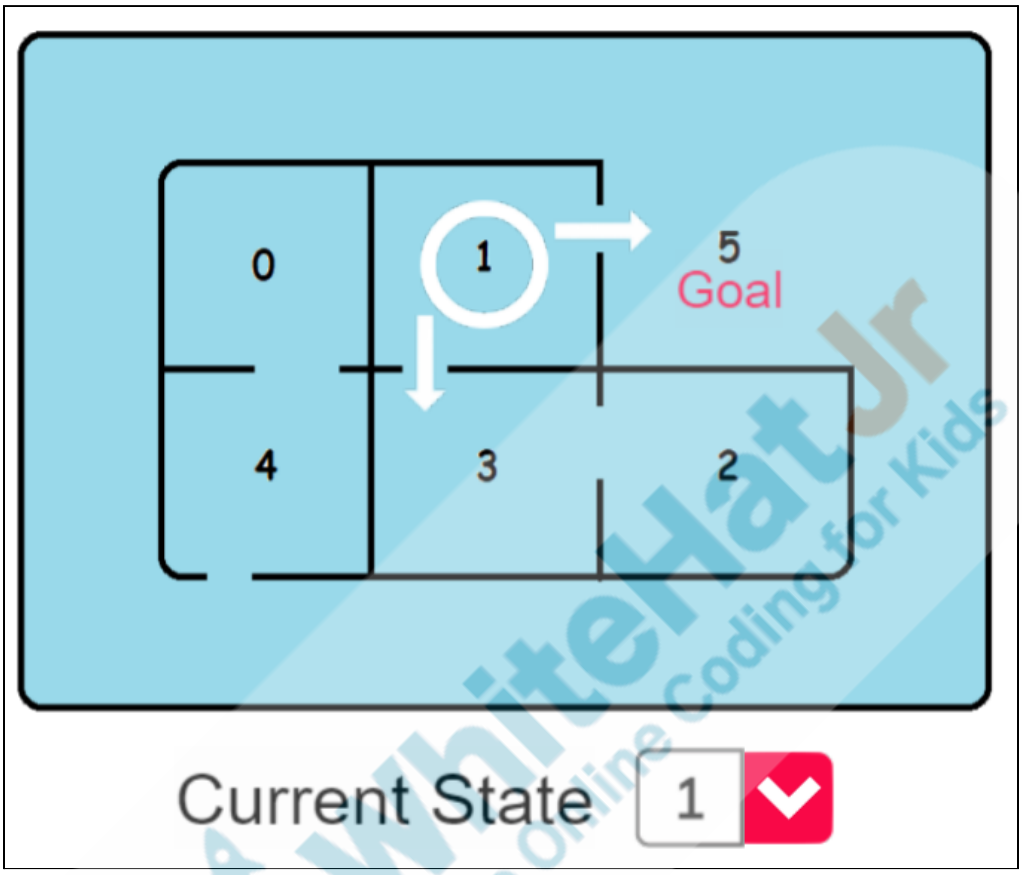
| | |
|---|---|
| • **Can't Move:** If there is no direct way from one room to another, then the reward is **-1.**<br><br>• **Move:** If the machine can move from the current room (state) to the next room (action) then the reward is **0.**<br><br>• **Goal:** If the machine is at/reached the goal, then the reward is **100.** | |
| **Creating a matrix of reward values:**<br><br>*Open Teacher Activity 1 to show the creations of the Reward Matrix while explaining it to the student.*<br><br>For possible states and actions:<br><br>• **Rows:** Represents the **States** of the Environment.<br>• **Columns:** Represents the **Actions** be taken by the Agent. | |




Let's start creating from state 1.

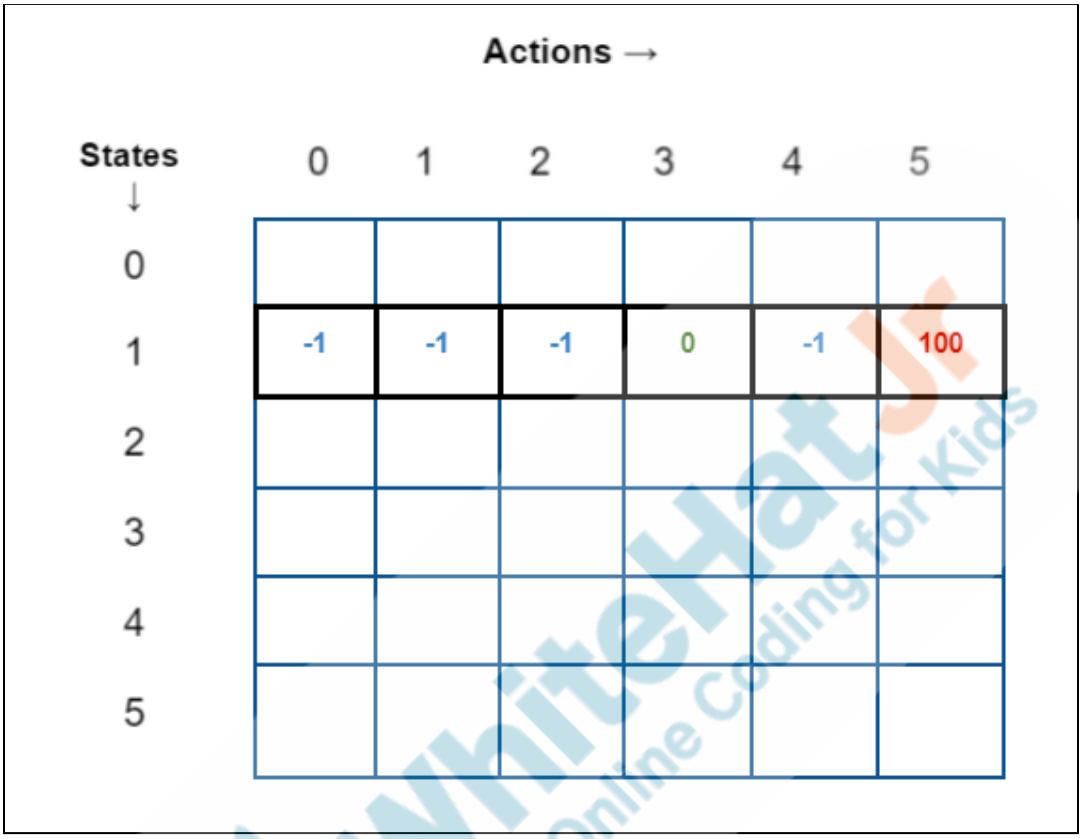
The machine can start from any state randomly, so we'll have to consider all the possibilities.

Let's fill in the matrix for this state.



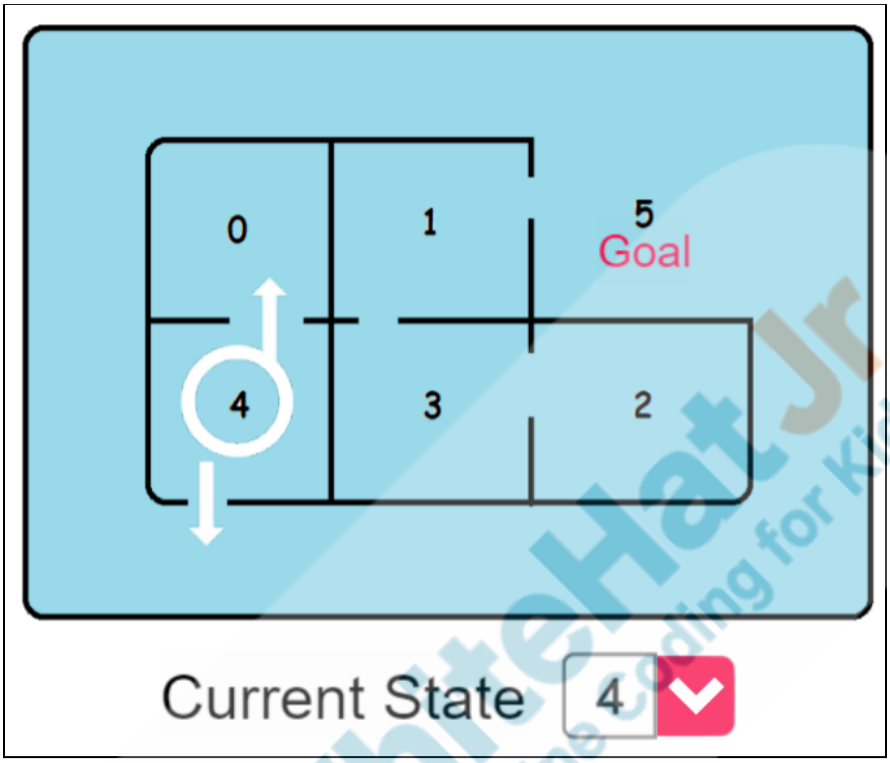

**Reward Matrix for State 1**:

- From state 1 it **can't move** to state numbers 0,1, 2, 4. So it'll be rewarded with **-1**.
- It can **move** to state 3 but to reach the goal it has to take more action. Thus it is rewarded as **0**.
- If it moves towards 5, it's the **Goal** so rewarded with **100**.

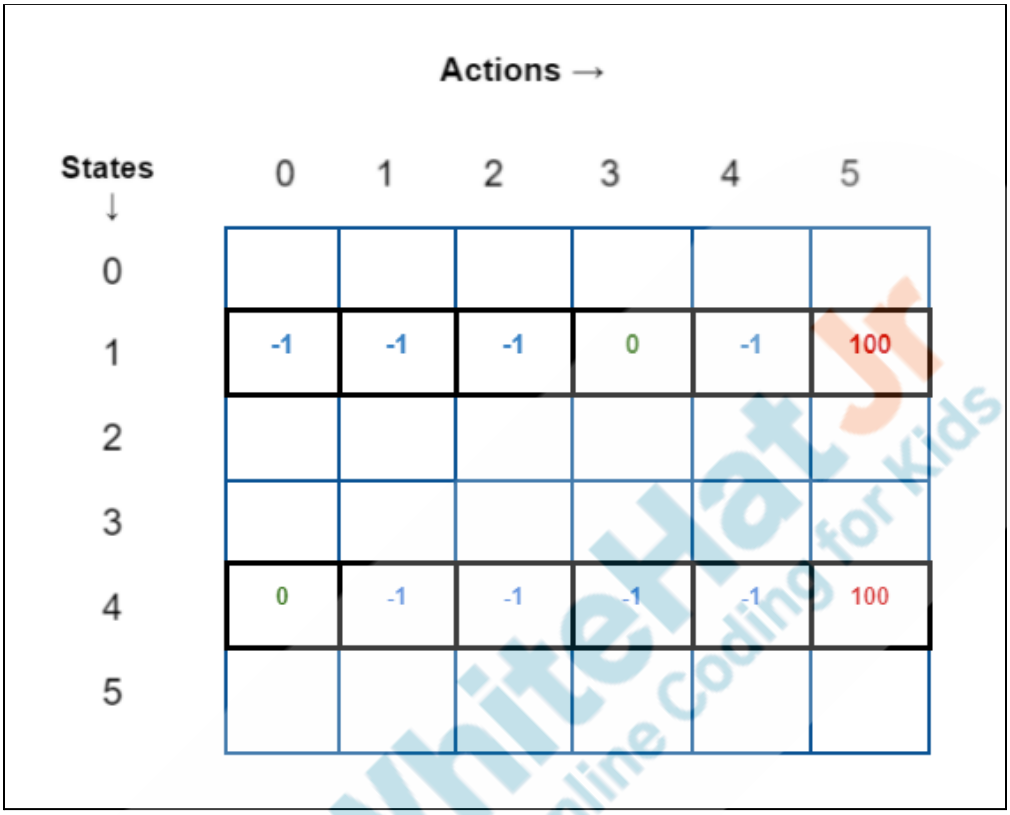



Let's try with state 4 as well.

From 4 there are two ways to take action.

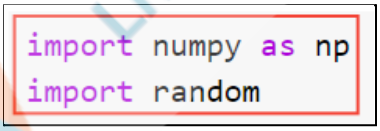



| | |
|---|---|
| Now you guide me to fill the matrix for all the possibilities for this state.<br><br>Great! | **ESR:** From state 4 to state 0 it'll get 0. For state 1,3,2 it'll be getting -1 and for state 5, it'll get 100. |




So this way we'll get the matrix as follows.


| | |
|---|---|
| So, let's start writing the code to create the reward matrix and write a logic to choose random action.<br><br>Are you excited? | **ESR:** Yes |
| **Teacher Stops Screen Share** | |
| So now it's your turn.<br>Please share your screen with me. | |
| **Teacher Starts Slideshow**<br>**Slide 15 to 18**<br>Refer to speaker notes and follow the instructions on each slide. | |
| We have one more class challenge for you.<br>Can you solve it? | |

| | |
|---|---|
| Let's try. I will guide you through it. | |
| **Teacher Ends Slideshow**  | |
| **STUDENT-LED ACTIVITY - 20 mins** | |
| • **Ask the student to press the ESC key** to come back to the panel.<br>• **Guide the student to start Screen Share.**<br>• **The teacher gets into Full Screen.** | |
| **Student Initiates Screen Share** | |
| **ACTIVITY**<br>• **Code to create reward matrix**<br>• **Choose random action** | |
| **Teacher Action** | **Student Action** |
| So, let's begin with creating a new notebook in Google Colab. We'll need to create a matrix consisting of arrays. We'll start by importing NumPy. Also to choose the state and action randomly, we'll import the random library. | |
|  | |
| Depending upon the states and action we had already seen how to create the reward matrix.<br>1. Create the reward matrix. It'll be a 2D NumPy array. Name it as **rewards**. | |

```
rewards = np.array([
    [-1, -1, -1, -1,  0,  -1],
    [-1, -1, -1,  0, -1, 100],
    [-1, -1, -1 , 0 ,-1 , -1],
    [-1,  0,  0 ,-1  ,-1 , -1],
    [0, -1, -1 , -1 ,-1 ,100],
    [-1, -1, -1, -1,  0, 100]
])
```

2. Create a function to set the initial state. As the initial state will be selected randomly using the **randint()** function of the random library.

```
def set_initial_state():
    return np.random.randint(0, 6)
```

3. Call the **set_initial_state()** function, each time it will return a random value of the state.

```
set_initial_state()
```

```
4
```

4. Next, create a function that takes the value of the current state of the agent and depending upon the reward matrix it should return the available actions.

```
def get_action(current_state, reward_matrix):
```

5. Define an empty list of available actions for the current state. First, we'll print the reward matrix.
6. Use a **for** loop to iterate through the row of the current state. Use an **enumerate()** function that

gives the index and the element present at that index. Suppose the current state is 3, the row corresponding to the state is **[-1 0 0 -1 -1 -1]**. The **enumerate()** function will return the list of tuples of index and elements. So the output (list of actions with rewards) will be:
**[(0,-1),(1,0),(2,0),(3,-1),(4,-1),(5,-1)].**

7. Now, Select the actions with positive rewards. Thus, if the reward is equal to **0** or **100**, the corresponding action will be appended to the list named **available_action.**

```
def get_action(current_state, reward_matrix):
    available_action = []
    print("reward_matrix","\n",reward_matrix)
    for action in enumerate(reward_matrix[current_state]):
        if action[1]!= -1:
            available_action.append(action[0])
```

8. Choose the action only amongst the actions with positive rewards.
9. Finally, print the available actions and the action chosen.

```
def get_action(current_state, reward_matrix):
    available_action = []
    print("reward_matrix","\n",reward_matrix)
    for action in enumerate(reward_matrix[current_state]):
        if action[1]!= -1:
            available_action.append(action[0])
    choose_action = random.choice(available_action)
    print("Random choice of action from",available_action,"is", choose_action)
    return choose_action
```

10. Call the **get_action()** function now by defining a current state.

```
current_state = 3
action = get_action(current_state, rewards)
print(action)
```

11. Run the program to check the output.

```
reward_matrix
 [[ -1  -1  -1  -1   0  -1]
 [ -1  -1  -1   0  -1 100]
 [ -1  -1  -1   0  -1  -1]
 [ -1   0   0  -1  -1  -1]
 [  0  -1  -1  -1  -1 100]
 [ -1  -1  -1  -1   0 100]]
Random choice of action from [1, 2] is 1
1
```

Great!!

We can try with another state and check the result. Let's try with state **4**.

```
current_state = 4
action = get_action(current_state, rewards)
print(action)
```

```
reward_matrix
 [[ -1  -1  -1  -1   0  -1]
 [ -1  -1  -1   0  -1 100]
 [ -1  -1  -1   0  -1  -1]
 [ -1   0   0  -1  -1  -1]
 [  0  -1  -1  -1  -1 100]
 [ -1  -1  -1  -1   0 100]]
Random choice of action from [0, 5] is 0
0
```

***Note:*** *In the above snippet of code if the student is not getting* ***0*** *as* ***action*** *make them rerun the cell. This way they will get 0.*

So you can the available actions are 0 and 5. It has chosen action 0 as the next move.

But don't you think that the action should be 5 as it is the goal. For this, we need to find a solution.

Great!!! we are able to create the reward matrix and depending upon this matrix the agent or machine will choose the actions. The actions with negative rewards are avoided and only with positive rewards are taken.

**Teacher Guides Student to Stop Screen Share**

**WRAP-UP SESSION - 05 mins**

**Teacher Starts Slideshow**
**Slide 19 to 24**

**Activity details**

**Following are the WRAP-UP session deliverables:**

- Appreciate the student.
- Revise the current class activities.
- Discuss the quizzes.

**WRAP-UP QUIZ**
Click on In-Class Quiz

**Continue WRAP-UP Session**
**Slide 25 to 30**

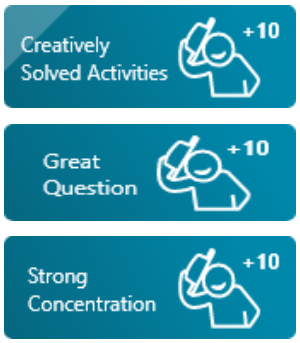
**Activity Details**

**Following are the session deliverables:**

- Explain the facts and trivia
- Next class challenge
- Project for the day
- Additional Activity (Optional)

**FEEDBACK**

- **Appreciate and compliment the student for trying to learn a difficult concept.**
- **Get to know how they are feeling after the session.**
- **Review and check their understanding.**

| Teacher Action | Student Action |
|---|---|
| You get “hats-off” for your excellent work!<br><br>In the next class, we’ll be learning to perform different operations on 1D and 2D arrays. | *Make sure you have given at least 2 hats-off during the class for:*<br><br>  |

**PROJECT OVERVIEW DISCUSSION**

Refer the document below in Activity Links Sections

**Teacher Clicks** ✖ End Class

| ACTIVITY LINKS | | |
|---|---|---|
| **Activity Name** | **Description** | **Links** |
| Teacher Activity 1 | Reward matrix creation | https://whitehatjrcontent.s3.ap-south-1.amazonaws.com/Teacher-Resources/COCOS_Applets/POC/Coding/SimpleQ-RL/rewardsOnly/index.html |
| Teacher Activity 2 | Reference Code | https://colab.research.google.com/drive/1MbRRulO1qM_w7UwdF7CtgzK6E3FWUnhJ?usp=sharing |
| Teacher Reference 1 | Project | https://s3-whjr-curriculum-uploads.whjr.online/a38e0057-5ac6-4f57-b7e1-dff01ea0e12d.pdf |
| Teacher Reference 2 | Project Solution | https://colab.research.google.com/drive/148QOe1D_ITfSAY5gLWXgTCfx5yu39UyJ?usp=sharing |
| Teacher Reference 3 | Visual-Aid | https://s3-whjr-curriculum-uploads.whjr.online/63674b23-ee98-40e9-a743-387cf2e1d939.html |
| Teacher Reference 4 | In-Class Quiz | https://s3-whjr-curriculum-uploads.whjr.online/4a3a35e0-29a7-4649-ae6a-68c40cf4af02.pdf |