Computer Science Demo Day

Welcome to Nanjing No. 13 Middle School's Open House Day!

We're so excited to have you visit our school and learn about the exciting things we teach here.

In this demo, you will learn a bit of computer science through hands-on projects!

You will learn how to code your own website using HTML.

If you want more of a challenge, you can make your own Tic Tac Toe game using HTML, CSS, and JavaScript.

If you like games, we made a fun guessing game. Can you find the best strategy?

Get ready for an exciting day to learn about the world of computers!

Make a Website

In this demo, you will learn how to customize your own website using HTML.

We already have most of the code ready for you but we want you to finish it!

Instructions:

- To get started, go to the Homepage on the website
- Then click on "Make a Website"
- Once you are there, go back to Visual Studio Code application
- Go to createWebsite → **index.html** (**not** ._index.html)

Since you may be new to coding, we will help you get started.

Important: After each code change, be sure to save the file using Ctrl (控制) + S

Adding a cloud picture:

- To add a picture, we use the img tag: **<img class="" src="">**
- The src allows the computer to find a picture and show it on the screen
- A class is just a name so we know how this **<img>** is different from others
- To start, simply write: **<img class="" src="">**
  - For class, write **class="top_cloud"**
  - For src, write **src="./images/cloud.png"**

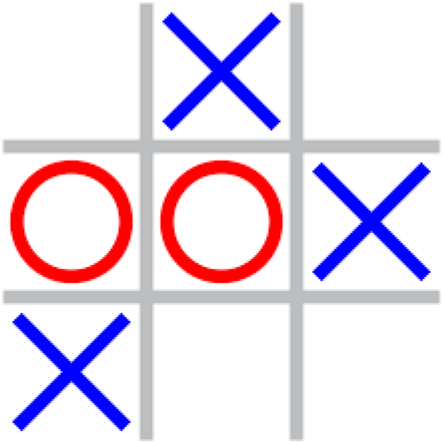
To say "Hello" or add your own message, simply replace [] with your own message.
For example:
[Say Hello or put your own message] → Hello! Welcome to my website!

Follow the rest of the instructions on the code that looks like <!- - instruction - - >

Make a Game

In this section, we’re going to make a game of Tic Tac Toe.

For those unfamiliar with the game, it looks like this:



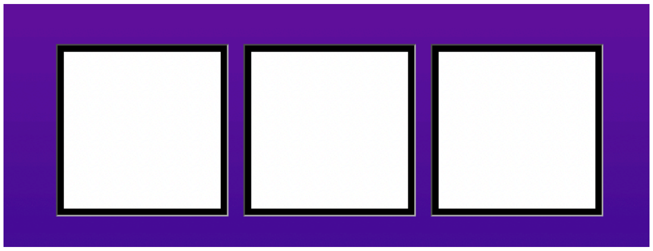
At the end of this, you will make a game AND be able to play with your friends and family!

Instructions:

- On the homepage, click on “Make a Game”
- Once you finish reading the starting message, click Start

HTML

Let’s start with making the board. You should see three squares like this:



This will be the first row of the game board.

To make the rest of the two, head over to VSCode and do the following:

- Go to makeTicTacToe → **tic_tac_toe.html**
- Go to the instructions on the code

Copy and paste the following:

```
<div class="row" id="row_1">
     <img class="block col_1" alt="">
     <img class="block col_2" alt="">
     <img class="block col_3" alt="">
</div>
```

This code is already on VS Code.

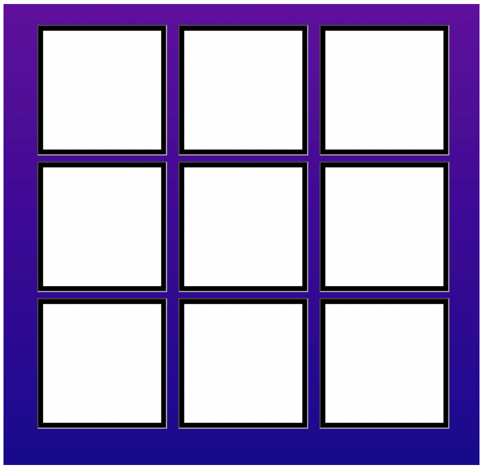
Once you do that, you should have:

```
<div class="row" id="row_1">
               <img class="block col_1" alt="">
               <img class="block col_2" alt="">
               <img class="block col_3" alt="">
</div>
<div class="row" id="row_2">
               <img class="block col_1" alt="">
               <img class="block col_2" alt="">
               <img class="block col_3" alt="">
</div>
<div class="row" id="row_3">
               <img class="block col_1" alt="">
               <img class="block col_2" alt="">
               <img class="block col_3" alt="">
</div>
```

Important: Change the **id=”row_2”** and **id=”row_3”**

Don’t forget to save!

Once you do that, you should see:



Great! You made the board!

<u>JavaScript</u>
Now, let’s get to the fun part. You’re going to let users play the game.

First, go to: makeTicTacToe → javascript → rules.js

You will see that much of the code has already been written for you. You just need to finish it.

To test, if you go back to the website and click on the squares, you will see that nothing happens.

That’s because we have not coded the rules of the game.

First, let’s start with letting the players put something on the board whenever they click on it.

Go to the following function:

```
function validTurn(block) {
   // when the player clicks on a square, a picture appears
   // make sure that it's the picture of their turn (Player 1 has X and Player 2 has
O)
}
```

This function takes in a `block` object as a variable. Don’t worry about what those mean right now. All it means is that when the user clicks on a square, the code will see it.

We just need to write the program so that the code knows what to do when the user clicks on a square.

If the square is empty, we can put an X or an O.

To write that, we can say: if square is empty → put X or O

It should look like this:

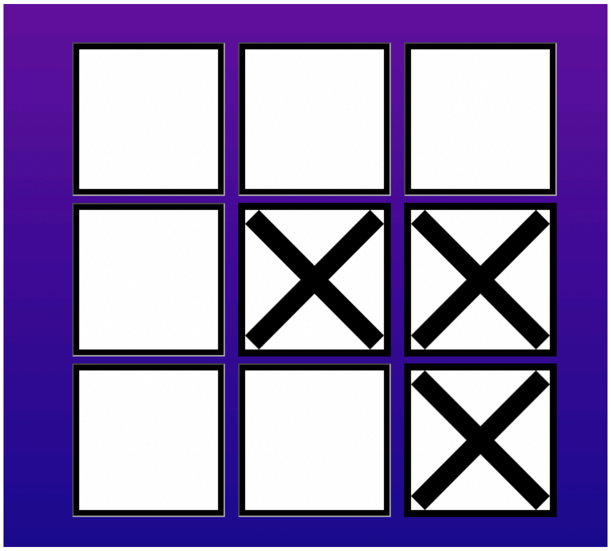
```
if(block.src == "") {
       block.src = `./assets/images/${turn}_mark.png`;
       // only switch turns when you can make the move
   }
```

This just means, if the square doesn’t have a picture, get the picture from the images folder and put it in the <img>.

Your final function should look like this:

```
function validTurn(block) {
   // square must be empty
   if(block.src == "") {
       block.src = `./assets/images/${turn}_mark.png`;
       // only switch turns when you can make the move
   }
}
```

Now, when you click on a square, you should see:



Great!
But there’s one problem, we have to switch between X and O for players to play.

Luckily, we can do that easily.

In the following function, we can change between X and O:

```
function changeTurn() {
   // each time the player goes, the turn switches from X to O and from O to X
}
```

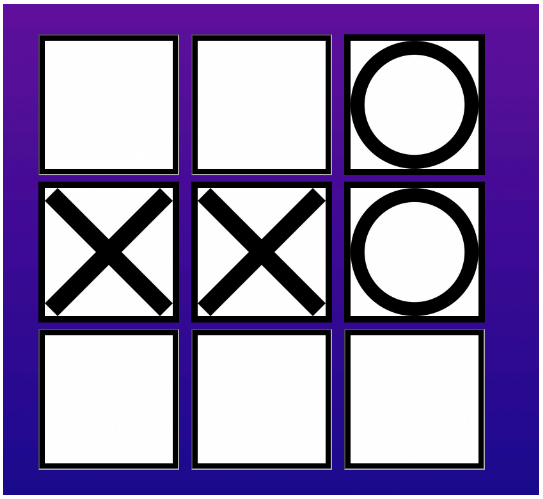
Challenge: Can you write the if-statement to complete this code?

Here’s how to write an if-statement in JavaScript:

```
if(this is true) {
       [do this];
   } else {
       [do this instead];
   }
```

Hint: use the `turn = "X"` variable

Once you complete that, you should have this:



Congrats! You’re almost done!

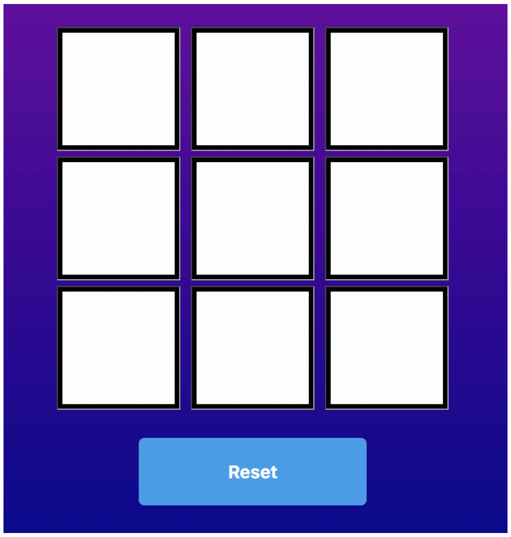
Now, we just need something that lets us reset the game whenever we want to play again.

Jump to the **tic_tac_toe.html** file and let’s put a reset button.

Underneath the instructions to add a button, simply add this code:

```
<div class="button_container">
     <button id="reset" class="cta">Reset</button>
</div>
```

Once added, this is what you should see:



Afterwards, go back to the **rules.js** file and add the following under the instructions to add a reset button:

```
document.getElementById("reset").onclick = reset;
```