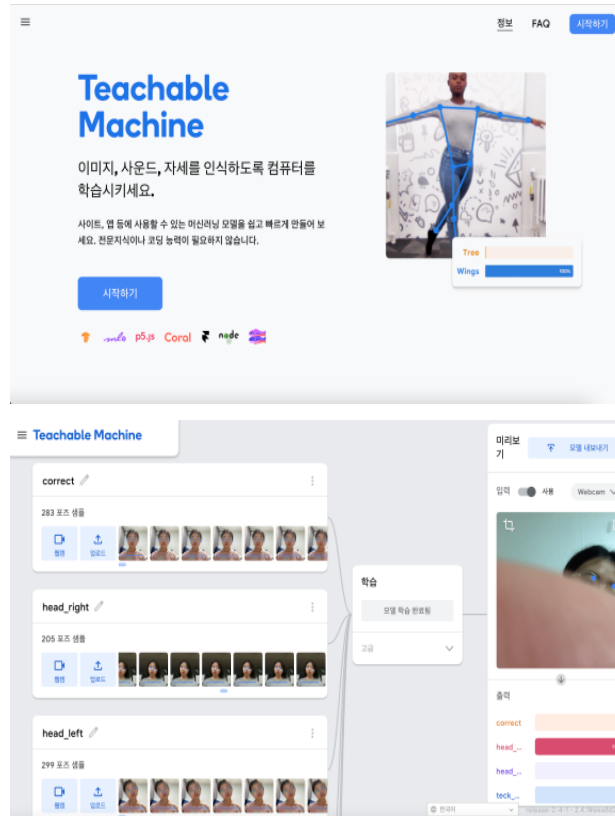


# 티처블머신 학습 결과 홈페이지 만들기



1. 티처블머신에서 프로젝트를 생성 후, 이미지나 음성 또는 포즈를 학습합니다.



2. '내려받기' 옆의 '모델 내려받기' 버튼을 클릭 후, '모델 업로드' 버튼을 눌러줍니다.

프로젝트에서 모델을 사용하려면 모델을 내보내세요. ✕

Tensorflow.js ⓘ

모델 내보내기:

☒ 업로드(공유 가능한 링크) ☐ 다운로드

공유 가능한 링크:

[https://teachablemachine.withgoogle.com/models/K\\_njbSr-S/](https://teachablemachine.withgoogle.com/models/K_njbSr-S/) 복사

모델을 업로드하면 Teachable Machine에서 이 링크에 모델을 무료로 호스팅합니다. (FAQ: 내 모델을 사용할 수 있는 사용자는 누구인가요?)

✓ 클라우드 모델이 최신 상태입니다.

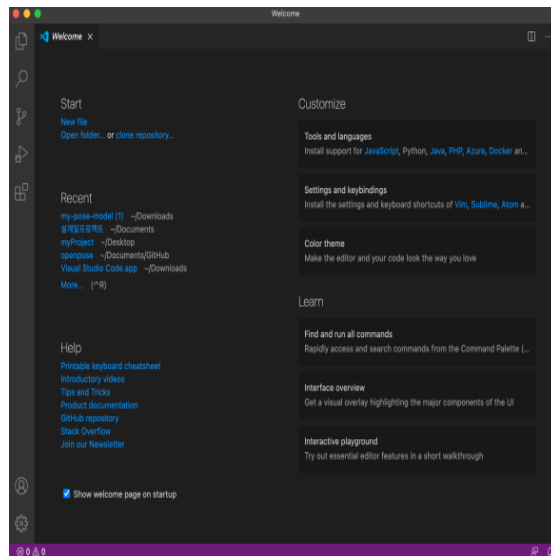
모델에서 사용할 코드 스니펫:

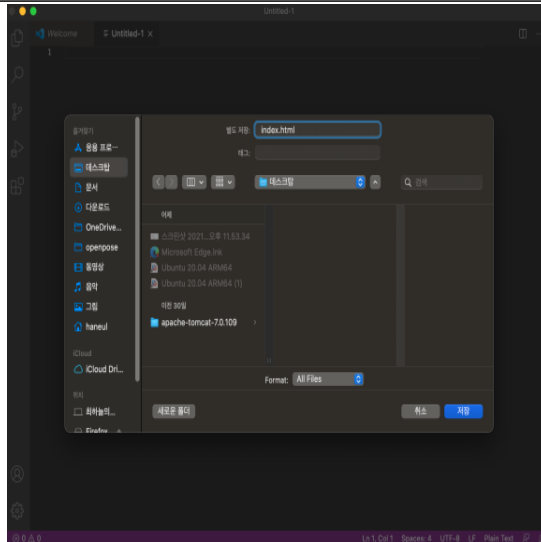
[Javascript](#) Github에 참여

Learn more about how to use the code snippet on [github](#).

```
<div>Teachable Machine Pose Model</div>
<button type="button" onclick="init()">Start</button>
<div><canvas id="canvas"></canvas></div>
<div id="label-container"></div>
<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@1.3.1/dist/tf.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/@teachablemachine/pose@0.8/dist/teachablemachine-pose.min.js"></script>
<script>
  const human = TeachableMachinePoseModel;
  const canvas = document.getElementById("canvas");
  const labelContainer = document.getElementById("label-container");
  const startButton = document.getElementById("start-button");
  startButton.addEventListener("click", () => {
    init();
  });
  function init() {
    const model = new human({
      url: "https://teachablemachine.withgoogle.com/models/K_njbSr-S/",
    });
    model.load().then(() => {
      model.run(canvas, labelContainer);
    });
  }
</script>
```

3. 공유 가능한 링크에 링크가 생성된 것을 확인합니다.



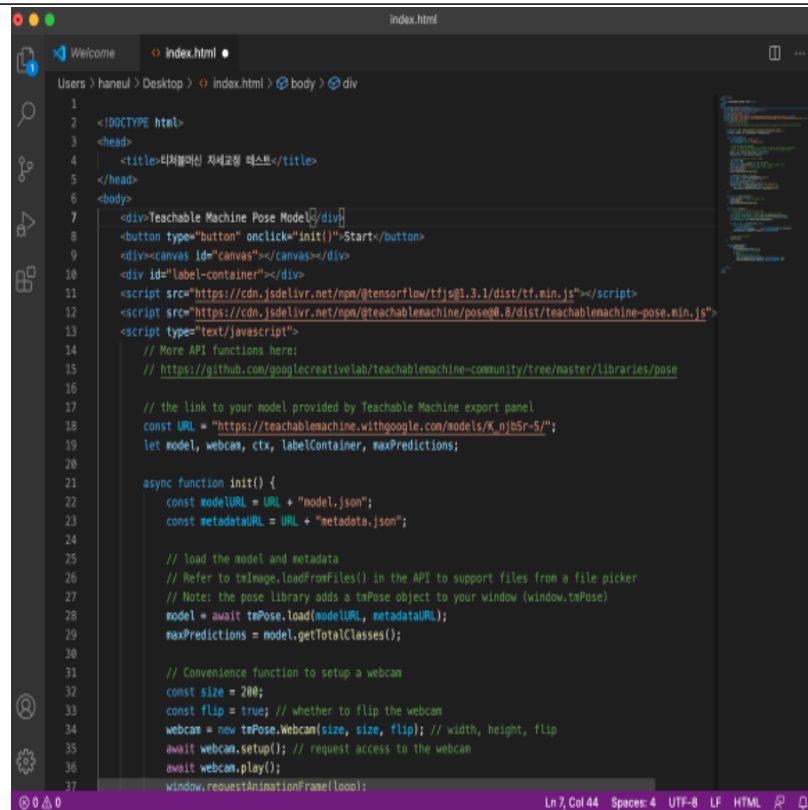


4. 비주얼스튜디오 코드에 index.html 파일을 생성합니다.  
그리고 파일에 코드를 붙여 넣어줍니다.

```
<!DOCTYPEhtml>
<head>
  <title>티쳐블머신 자세교정 테스트</title>
</head>
<body>
</body>
</html>
```



5. JavaScript 코드를 복사합니다.



```
1 <!DOCTYPE html>
2 <head>
3   <title>티쳐블머신 자세교정 테스트</title>
4 </head>
5 <body>
6   <div>Teachable Machine Pose Model</div>
7   <button type="button" onclick="init()">Start</button>
8   <div><canvas id="canvas"></canvas></div>
9   <div id="label-container"></div>
10  <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@1.3.1/dist/tf.min.js"></script>
11  <script src="https://cdn.jsdelivr.net/npm/@teachablemachine/pose@0.8/dist/teachablemachine-pose.min.js"></script>
12  <script type="text/javascript">
13    // More API functions here:
14    // https://github.com/googlecreativelab/teachablemachine-community/tree/master/libraries/pose
15
16    // the link to your model provided by Teachable Machine export panel
17    const URL = "https://teachablemachine.withgoogle.com/models/K_njbSr-S/";
18    let model, webcam, ctx, labelContainer, maxPredictions;
19
20    async function init() {
21      const modelURL = URL + "model.json";
22      const metadataURL = URL + "metadata.json";
23
24      // load the model and metadata
25      // Refer to tmPose.loadFromFiles() in the API to support files from a file picker
26      // Note: the pose library adds a tmPose object to your window (window.tmPose)
27      model = await tmPose.load(modelURL, metadataURL);
28      maxPredictions = model.getTotalClasses();
29
30      // Convenience function to setup a webcam
31      const size = 200;
32      const flip = true; // whether to flip the webcam
33      webcam = new tmPose.Webcam(size, size, flip); // width, height, flip
34      await webcam.setup(); // request access to the webcam
35      await webcam.play();
36      window.requestAnimationFrame(loop);
37    }
```

6. body 태그 사이에 붙여 넣어줍니다.  
붙여 넣어준 코드는 다음과 같습니다.

```
<!DOCTYPE html>
<head>
  <title>티쳐블머신 자세교정 테스트</title>
</head>
<body>
  <div>Teachable Machine Pose Model</div>
  <button type="button" onclick="init()">Start</button>
  <div><canvas id="canvas"></canvas></div>
  <div id="label-container"></div>
  <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@1.3.1/dist/tf.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/@teachablemachine/pose@0.8/dist/teachablemachine-pose.min.js"></script>
  <script type="text/javascript">
    // More API functions here:
    /
    https://github.com/googlecreativelab/teachablemachine-community/tree/master/libraries/pose

    // the link to your model provided by Teachable Machine export panel
    const URL="https://teachablemachine.withgoogle.com/models/K_njbSr-S/";
    let model, webcam, ctx, labelContainer, maxPredictions;

    async function init() {
      const modelURLURL"model.json";
      const metadataURLURL"metadata.json";
```

```

// load the model and metadata
// Refer to tmImage.loadFromFiles() in the API to support files from a file picker
// Note: the pose library adds a tmPose object to your window (window.tmPose)
modelawait tmPose.load(modelURL, metadataURL);
maxPredictionsmodel.getTotalClasses();

// Convenience function to setup a webcam
const size200;
const fliptrue; // whether to flip the webcam
webcamnew tmPose.Webcam(size, size, flip); // width, height, flip
await webcam.setup(); // request access to the webcam
await webcam.play();
window.requestAnimationFrame(loop);

// append/get elements to the DOM
const canvasdocument.getElementById("canvas");
canvas.widthsize; canvas.heightsize;
ctxcanvas.getContext("2d");
labelContainerdocument.getElementById("label-container");
for(let i0; i < maxPredictions; i++) { // and class labels
labelContainer.appendChild(document.createElement("div"));
}
}

async function loop(timestamp) {
webcam.update(); // update the webcam frame
await predict();
window.requestAnimationFrame(loop);
}

async function predict() {
// Prediction #1: run input through posenet
// estimatePose can take in an image, video or canvas html element
constpose, posenetOutput} = await model.estimatePose(webcam.canvas);
// Prediction 2: run input through teachable machine classification model
const predictionawait model.predict(posenetOutput);

for(let i0; i < maxPredictions; i++) {
const classPrediction=
prediction[i].className": "prediction[i].probability.toFixed(2);
labelContainer.childNodes[i].innerHTMLclassPrediction;
}

// finally draw the poses
drawPose(pose);
}

function drawPose(pose) {
if(webcam.canvas) {
ctx.drawImage(webcam.canvas, 0, 0);
// draw the keypoints and skeleton
if(pose) {
const minPartConfidence0.5;
tmPose.drawKeypoints(pose.keypoints, minPartConfidence, ctx);
tmPose.drawSkeleton(pose.keypoints, minPartConfidence, ctx);
}
}
}
</script>

```

</body>  
</html>

프로젝트에서 모델을 사용하려면 모델을 내보내세요.

Tensorflow.js

모델 내보내기:

☒ 업로드(공유 가능한 링크) ☐ 다운로드

공유 가능한 링크:

[https://teachablemachine.withgoogle.com/models/K\\_njbSr-S/](https://teachablemachine.withgoogle.com/models/K_njbSr-S/) 복사

모델을 업로드하면 Teachable Machine에서 이 링크에 모델을 무료로 호스팅합니다. (FAQ: 내 모델을 사용할 수 있는 사용자는 누구인가요?)

✓ 클라우드 모델이 최신 상태입니다.

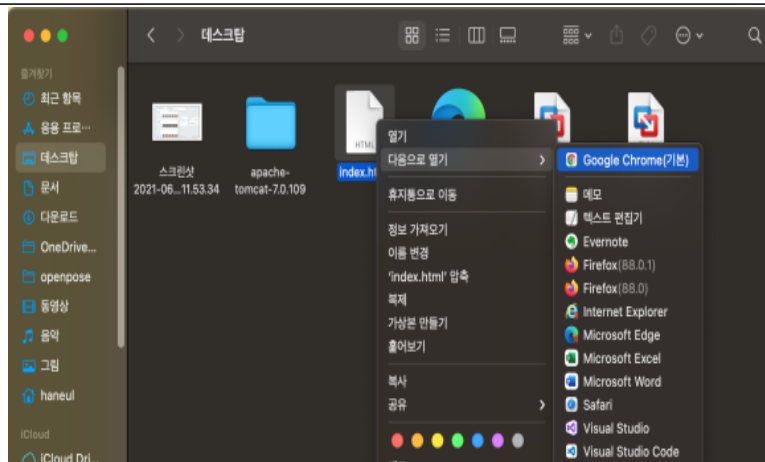
모델에서 사용할 코드 스니펫:

Javascript [Github에 참여](#)

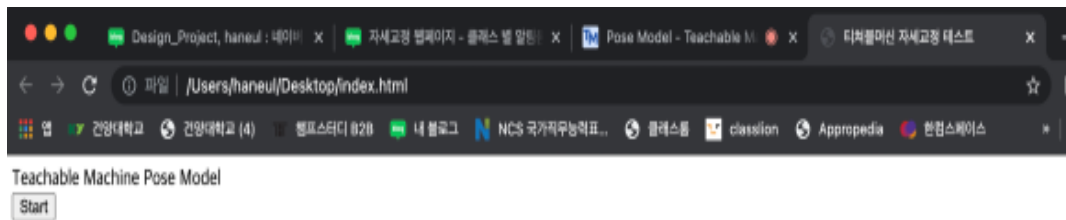
Learn more about how to use the code snippet on [github](#).

7. 공유 가능한 링크를 복사하여, 다음 자리에 붙여 넣어 줍니다.

```
<script type="text/javascript">
// More API functions here:
//https://github.com/googlecreativelab/teachablemachine-community/tree/master/libraries/pose
// the link to your model provided by Teachable Machine export panel
const URL = "공유 가능한 링크";
letmodel,webcam,ctx,labelContainer,maxPredictions;
async function init() {
constmodelURL = URL + "model.json";
constmetadataURL = URL + "metadata.json";
```



8. 다음과 같이 열어줍니다.



9. Start를 누르면 내가 생성한 프로젝트의 학습 결과가 활성화됩니다.