Readme.md

Chatbots are showing up everywhere, but how do they work? Some chatbots rely on artificial intelligence and machine learning, but some chatbots follow a simple script. In this project try out a simple chatbot in Thimble and create your own remix!

Version 2.3

BotChat.js

// botChat is the function that picks the bot's next message.

function botChat() {

// Set the bot as the sender of the next message.

nextMessage.sender = "bot";

if (botCount >= botScript.length) {

nextMessage.message = goodbye

botSilent = true;

} else {

// Set the bot's next message as the next string in the botScript array.

nextMessage.message = botScript[botCount];

}

// Check sendSpecialChat to see if anything special should happen.

if (sendSpecialChat[0]) {

nextMessage.message = sendSpecialChat[1];

}

// Send the bot's message.

send(nextMessage.sender, nextMessage.message);

// Count 1 more chat that the bot has sent unless the chat was a sendSpecialChat.

if (sendSpecialChat[0]) {

sendSpecialChat = [false, ""];

} else {

botCount += 1;

}

// Start listening again after the bot has sent a message.

listenFor();

}

// The comment below tells this file about variables we define in other files.

/\*

global botCount

global botScript

global nextMessage

global botSilent

global goodbye

global sendSpecialChat

global send

global listenFor

\*/

ChatEngine.js

/\*

This JavaScript file has all of the code to find chats from the bot and user and create the chat interface. Because the chat window will change as you type we need to manipulate the DOM (Document Object Model) and add HTML to display on the screen.

\*/

function lookForChat() {

// If there have been no chats yet, start the bot.

if (count == 0) {

startBot();

}

console.log("🌀 Looking for chat " + (count + 1));

// check who sent the last chat

last = nextMessage.sender;

if (last == "bot") {

// if the bot chatted last wait for the user to send a chat

userChat();

} else {

// Send the cursor to the compose text area.

composer.focus();

// If botSilent is true the bot is done chatting

// Set nextMessage.sender to "bot" to make the user chat next

// Run listenFor() to wait for the user to chat.

if (botSilent) {

nextMessage.sender = "bot";

nextMessage.message = "";

listenFor();

} else {

// If the user chatted last or the chat just started have the bot send a chat.

// Set the appropriate wait time to make the bot feel realistic.

// Then run the botChat function which will find the right message for the bot

if (count == 0) {

wait = 100;

} else {

wait = 500;

}

setTimeout(function(){

console.log("🕙 Waiting for bot");

botChat();

}, wait);

}

}

}

// startBot is a function that starts the bot for the first time. It clears away the start button from the HTML.

function startBot(){

console.log("🤖 Starting bot . . .");

chatArea.innerHTML = '';

document.getElementById('compose-area').style.display = 'block';

}

// The comment below tells this file about variables we define in other files.

/\*

global count

global last

global nextMessage

global userChat

global composer

global botSilent

global listenFor

global wait

global botChat

global chatArea

\*/

Index.html

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>JavaScript Chatbot</title>

<link rel="stylesheet" href="style.css">

<link id="favicon" rel="icon" href="https://glitch.com/edit/favicon.ico" type="image/x-icon">

</head>

<body>

<div id="wrapper">

<div id="chat-area">

<div id="start-screen">

<div id="startButton">

<div id="robot">

<img src="https://cdn.glitch.com/e5ca6c0c-b21f-4a54-a1ad-6285b1bf8390%2Frobot.svg">

</div>

<img src="https://cdn.glitch.com/e5ca6c0c-b21f-4a54-a1ad-6285b1bf8390%2Fpointer.png" id="cursor"> or ⏎ to start!

</div>

</div>

</div>

<div id="compose-area">

<textarea id="composer" class="text-input" type="text" autofocus="autofocus"></textarea>

<div id="sendButton">

<button>&nbsp;↵&nbsp;</button>

</div>

</div>

</div>

<script src="variables.js"></script>

<script src="listener.js"></script>

<script src="thingsTheBotSays.js"></script>

<script src="chatEngine.js"></script>

<script src="botChat.js"></script>

<script src="userChat.js"></script>

<script src="send.js"></script>

<div class="glitchButton" style="position:fixed;top:20px;right:20px;"></div>

<script src="https://button.glitch.me/button.js"></script>

</body>

</html>

Listener.js

/\*

This JavaScript file has code to allow a user to choose to press the Start or ⏎ button, or to press the Enter key on their keyboard using the .keycode property.

\*/

// listener is a variable to decide what input to look for

var enterListener = window;

var startListener = document.getElementById("startButton");

var sendListener = document.getElementById("sendButton");

var composer = document.getElementById("composer");

// Listen for the enter key on the start screen to start the chat

if (count == 0) {

listenFor()

}

// A pair of functions that chain together to decide what part of the page to listen to, and then see if the 'enter' key is pressed. If it is, run the chat() function to submit an answer if there is one, and ask a new question.

function listenFor() {

console.log("🔈 Starting to listen for Enter Key or Mouse Click.");

enterListener.addEventListener("keydown", listen);

startListener.addEventListener("click", listen);

sendListener.addEventListener("click", listen);

}

function listen(e) {

if (e.keyCode === 13) {

e.preventDefault();

if (!e.shiftKey) {

console.log("⌨ Listen Event: " + e + ", Type: " + e.type + ", Keycode: " + e.keyCode);

pauseListening();

lookForChat();

}

}

if (e.type == "click") {

console.log("🖱 Listen Event: " + e + ", Type: " + e.type + ", Button: " + e.which);

pauseListening();

lookForChat();

}

}

// To avoid double submitting on enter key if someone clicks the button we have top stop listening for enter until the robot sends a question.

function pauseListening() {

enterListener.removeEventListener("keydown", listen);

startListener.removeEventListener("click", listen);

sendListener.removeEventListener("click", listen);

console.log("🔇 Stop listening for enter or click.");

}

// The comment below tells this file about variables we define in other files.

/\*

global count

global lookForChat

\*/

Send.js

// send is the function that sends the next message stored in nextMessage object.

function send(sender, message) {

console.log("🗨 " + sender + ": " + message);

// Insert the nextMessage into the HTML.

chatArea.insertAdjacentHTML("beforeend", "<div id='chat-" + count + "' class='chat-container'><div class='chat-wrapper' id='chat-a-" + count + "'><p id='a-' class='chat-" + sender + "'>" + message + "</p><div class='avatar avatar-" + sender + "'></div></div></div>");

// Scroll the most recent message onto the screen.

document.getElementById('chat-' + count).scrollIntoView();

// Count one more message that has been sent.

count += 1;

}

// The comment below tells this file about variables we define in other files.

/\*

global chatArea

global count

\*/

Style.css

// send is the function that sends the next message stored in nextMessage object.

function send(sender, message) {

console.log("🗨 " + sender + ": " + message);

// Insert the nextMessage into the HTML.

chatArea.insertAdjacentHTML("beforeend", "<div id='chat-" + count + "' class='chat-container'><div class='chat-wrapper' id='chat-a-" + count + "'><p id='a-' class='chat-" + sender + "'>" + message + "</p><div class='avatar avatar-" + sender + "'></div></div></div>");

// Scroll the most recent message onto the screen.

document.getElementById('chat-' + count).scrollIntoView();

// Count one more message that has been sent.

count += 1;

}

// The comment below tells this file about variables we define in other files.

/\*

global chatArea

global count

\*/

Thingsthebotsays.js

/\*

This JavaScript file has the robot's 'script', which is the list of questions or statements the robot will send as chat.

\*/

// botScript is a list of questions the robot will ask the user.

var botScript = [

"Hello! What is your name?",

"This is the second thing I say",

"This is the third thing I say",

"This is the last thing I say before I say goodbye"

];

// goodbye is a variable that stores what the robot will say when it runs out of other things to say.

var goodbye = "Goodbye! I have to go now.";

userchat.js

// userChat is the function that waits for the user to send a message.

function userChat() {

// Find where the user is inputing text.

compose\_area = document.getElementById('composer');

// Set the user as the sender of the next message.

nextMessage.sender = "user";

// Get the user's input in the compose\_area and clear the compose\_area.

nextMessage.message = compose\_area.value;

compose\_area.value = "";

// We need to convert the user's message to upper case to check if it matches with any prompts using the .toUpperCase() function.

uppercase = nextMessage.message.toUpperCase();

// We can test if the user's message matches any of the prompts using if statements.

console.log("✍ Code for custom prompts goes here");

if (uppercase == "HAPPY BIRTHDAY") {

sendSpecialChat = [true, "Thank you! How did you know it's my birthday?!"];

}

// sendSpecialChat is an array that will override the next thing the bot says with the second value if the first value is true. If the first value is false the bot will say the next thing in the script.

// Send user's message.

send(nextMessage.sender, nextMessage.message);

// Count 1 more chat that the user has sent.

userCount += 1;

// Ask the bot for another chat.

lookForChat()

}

// The comment below tells this file about variables we define in other files.

/\*

global compose\_area

global nextMessage

global uppercase

global sendSpecialChat

global send

global userCount

global lookForChat

\*/

Variables.js

/\*

In JavaScript we use variables to store information for later use. To allow variables to be used globally which preserves their information across many functions we have to declare them first with the 'var' keyword. In this JavaScript file we declare all the global variables we will need in other files.

\*/

// chatArea is the variable that stores the place on the screen the chats will appear.

var chatArea = document.getElementById('chat-area');

// count is a variable that stores how many total chats have been sent.

var count = 0;

// botCount and userCount are variables that store how many chats each the bot and the suer have sent.

var botCount = 0;

var userCount = 0;

// nextMessage is an object variable that stores the next message that will be sent and who will be sending it.

var nextMessage = {

message: "",

sender: ""

};

// sendSpecialChat is a variable that stores if the bot should say something off script, and what that should be.

var sendSpecialChat = [false, ""];

// botSilent is a variable that stores when the bot is done speaking because it has said all of the things in the script.

var botSilent = false;