

Summary

Title: 4 Temperaments Personality Quiz

This online quiz will sort the quiz taker into 4 personality temperaments: Sanguine, Choleric, Melancholic, or Phlegmatic. The user will answer 12 questions based on how strongly they agree or disagree with the question statement. Then, at the end, they will be sorted and given a longer description of the kind of person they are.

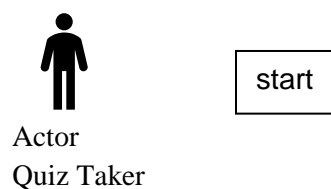
Intended user is anyone who likes to take personality quizzes

I've seen research done on the theory that every person can be categorized into four personality temperaments, but haven't actually seen a quiz determining it. I think people like simplicity (and Myer Briggs results, for example, can be difficult to parse through) so this quiz allows them to be sorted without all the details of other personality quizzes. (Besides that, there isn't really a huge problem that necessitates this project, it's more for fun).

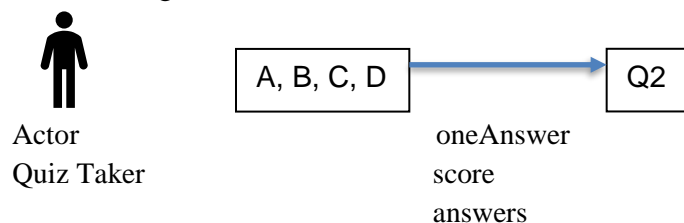
I will be requiring the use of HTML via Bottle (PythonAnywhere's web app functionality). No external downloads are necessary—the only thing needed to access the project is a web browser. Mobile use is supported, but the visual layout is not ideal.

Use Case Analysis

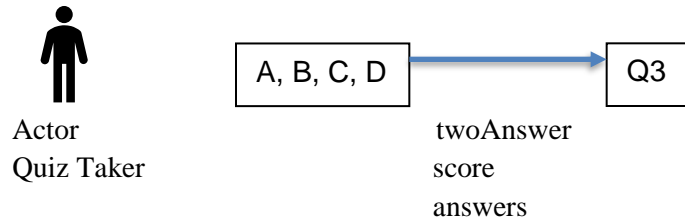
Screen 1 - Start



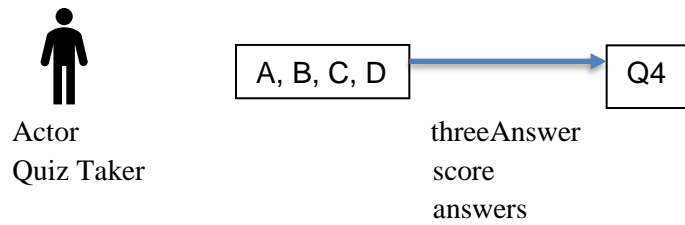
Screen 2 – Q1



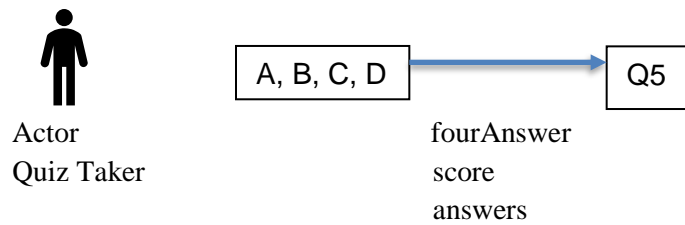
Screen 3 – Q2



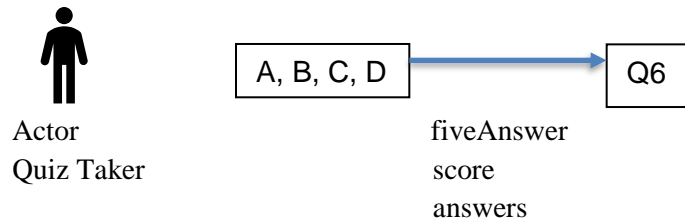
Screen 4 – Q3



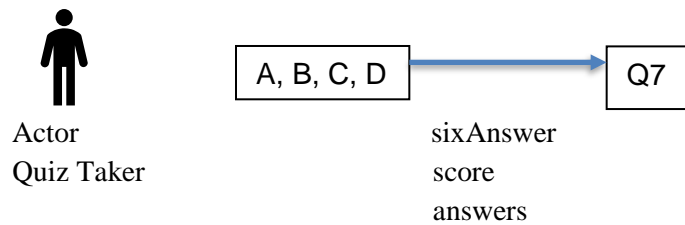
Screen 5 – Q4



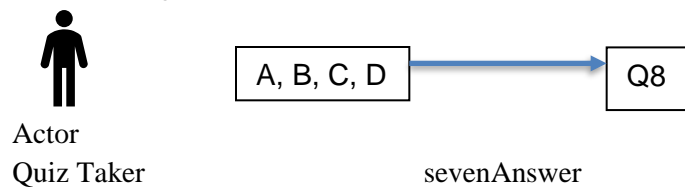
Screen 6 – Q5



Screen 7 – Q6



Screen 8 – Q7



Screen 9 – Q8



Actor
Quiz Taker



score
answers

eightAnswer
score
answers

Screen 10 – Q9



Actor
Quiz Taker



nineAnswer
score
answers

Screen 11 – Q10



Actor
Quiz Taker



tenAnswer
score
answers

Screen 12 – Q11



Actor
Quiz Taker



elevenAnswer
score
answers

Screen 13 – Q12



Actor
Quiz Taker



twelveAnswer
score
answers

Screen 14 – Result



Actor
Quiz Taker

Largest

Screen 15 (possible) – ones



Actor
Quiz Taker

No data, just description

Screen 15 (possible) – tens



Actor
Quiz Taker

No data, just description

Screen 15 (possible) – hundreds



Actor
Quiz Taker

No data, just description

Screen 15 (possible) – thousands



Actor
Quiz Taker

No data, just description

Data Design

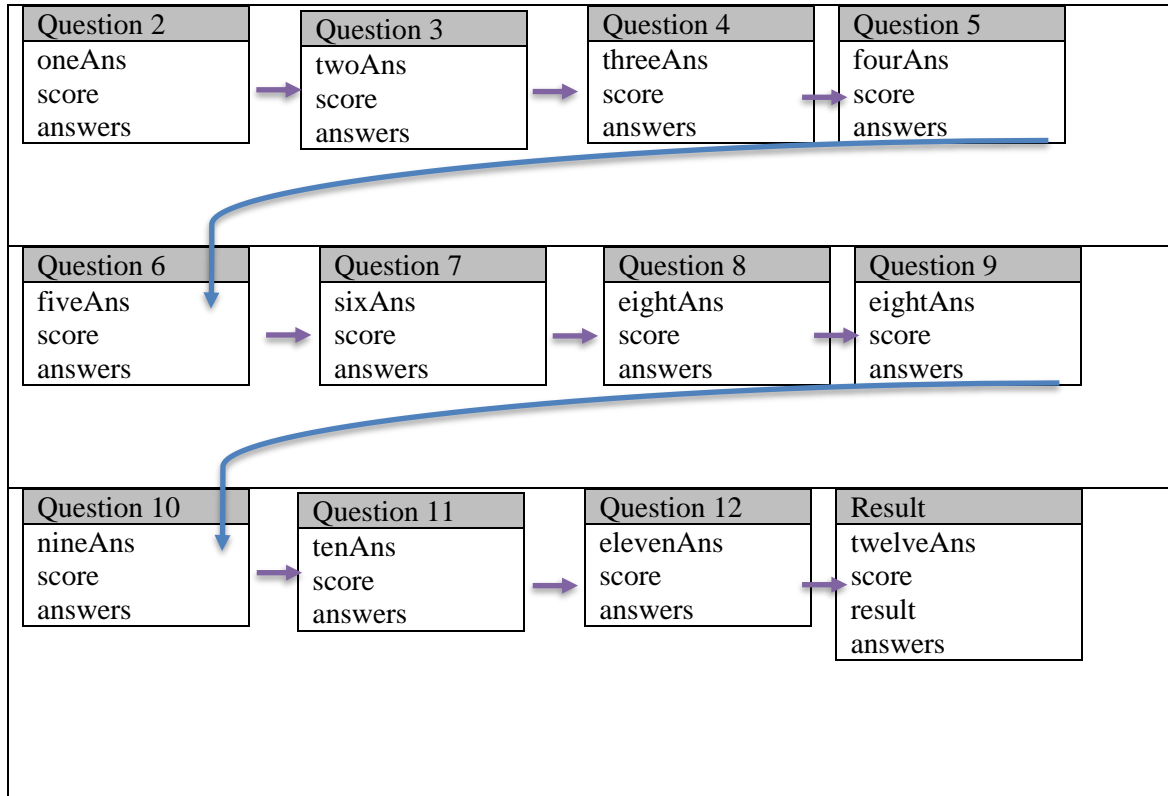
This program revolves around the integer “score” variable being properly updated throughout the quiz depending on what the user answers. Score will be incremented or decremented based on if the user agrees (increment) or disagrees (decrement) with a question statement.

The [questionNumber]Answer variables will be sent to the next page and will be the request of getting which radio button (A, B, C, or D) the user picked. Based on this, a conditional will be used to determine how to update the score variable.

There will also be a string ‘answers’ variable that is sent throughout the program. The values A, B, C, or D will be concatenated, plus a space, onto answers each page based on what the user answered for each

question. In the results page, this will then be converted into a list, split based on the space delimiter, and used for tie calculations.

The data will follow a chronological order with each piece being sent to the next page:



UI Design

All the pages in the quiz will have the same styling:

Cormorant Upright font

Light purple-pink gradient background color

Everything centered left

BUTTONS:

Each question will have a submit button with written word “next”

Each question will have 4 radio buttons, all named “[questionNumber]Answer”:

First, with the id “A” and label “Strongly Agree”

Second, with the id “B” and label “Agree”

Third, with the id “C” and label “Disagree:

Fourth, with the id “D” and label “Strongly Disagree”

Each radio button will be in its own paragraph

Each question will have a button with written word “back”

CSS Algorithm, in external style sheet named “quizStyle.css”:

In body:

Make the background gradient from light purple to pink
Re-size the background height to 250% of its original size

Have a class called biggerImg for the qResult page:
Make the background gradient from light purple to pink
Re-size the background height to 450% of its original size

In h1:
Make the font “Cormorant Upright”
Make the size of the header 37px

In p:
Make the font Cormorant Upright
Make the size 20px
Make the text color gray

In level 2 header:
Make the font Cormorant Upright
Make the text color gray

In button:
Make the font size 15px
Make the font “Cormorant Upright”
Make the button wide

In unordered list:
Make the font “Cormorant Upright”
Make the font size 20 px
Make the text color gray

Algorithm

Start class

|

Set up the doctype

Declare the html to be english

In the head container:

Make the title of the tab “Question1”

Make the charset UTF-8

add a link to quizStyle.css, rel "stylesheet" and type attribute "text/css"

have a reference link to the google fonts for "Cormorant Upright" and outline effect and relative attribute "stylesheet"

Make a header stating “4 Temperaments Personality Quiz”

Make a paragraph describing what the quiz is

Make a form with action attribute that goes to “/q2” and post method attribute:

Make a paragraph for submit button:

Button type is “submit” and the text will be “Start Quiz”

|

Q1 class

|

Set up the doctype

Declare the html to be english

In the head container:

Make the title of the tab “Question1”

Make the charset UTF-8

add a link to quizStyle.css, rel "stylesheet" and type attribute "text/css"

have a reference link to the google fonts for "Cormorant Upright" and outline effect and relative attribute "stylesheet"

In the body container:

Make a level 1 header saying “Question 1”

Make a paragraph asking the Saguine question “You can start a conversation with just about anyone”

Make a form with action attribute that goes to “/q2” and post method attribute:

Make a paragraph for first answer option:

Make an input of type radio:

name attribute is “oneAnswer”

id attribute is “A”

value attribute is also “A”

label attribute is “Strongly Agree”

Make a paragraph for second answer option:

Make an input of type radio:

name attribute is “oneAnswer”

id attribute is “B”

value attribute is also “B”

label attribute is “Agree”

Make a paragraph for third answer option:

Make an input of type radio:

name attribute is “oneAnswer”

id attribute is “C”

value attribute is also “C”

label attribute is “Disagree”

Make a paragraph for fourth answer option:

Make an input of type radio:

name attribute is “oneAnswer”

id attribute is “D”

value attribute is also "D"

label attribute is "Strongly Disagree"

Make a paragraph for submit button:

Button type is "submit" and the text will be "next"

Q2 class

Have a chunk of python code:

Import request library

Declare and initialize "score" variable to 3333

Declare and initialize "answers" variable to be an empty string

Request the "oneAnswer" from the previous page and set it equal to "oneAnswer"

If oneAnswer is equal to "A":

Increment score by 2000

Otherwise if oneAnswer is equal to "B":

Increment score by 1000

Otherwise if oneAnswer is equal to "C":

Decrement score by 1000

Otherwise if oneAnswer is equal to "D":

Decrement score by 2000

Otherwise, assuming there was no input:

Set oneAnswer equal to the string 0

End the python block

Concatenate "oneAnswer" and a space key to answers variable

End the python code

Set up the doctype

Declare the html to be english

In the head container:

Make the title of the tab "Question2"

Make the charset UTF-8

add a link to quizStyle.css, rel "stylesheet" and type attribute "text/css"

have a reference link to the google fonts for "Cormorant Upright" and outline effect and relative attribute "stylesheet"

In the body container:

Make a level 1 header saying "Question 2"

Make a paragraph asking the Melancholic question "You are an emotional person, prone to crying or laughing at the drop of a hat"

Make a form with action attribute that goes to "/q3" and post method attribute:

Make an input of type "hidden" that will send the data without the user seeing it:

Name attribute will be “score”

Value attribute will be the python value of “score”

Make an input of type “hidden” that will send the data without the user seeing it:

Name attribute will be “answers”

Value attribute will be the python value of “answers”

Make a paragraph for first answer option:

Make an input of type radio:

name attribute is “twoAnswer”

id attribute is “A”

value attribute is also “A”

label attribute is “Strongly Agree”

Make a paragraph for second answer option:

Make an input of type radio:

name attribute is “twoAnswer”

id attribute is “B”

value attribute is also “B”

label attribute is “Agree”

Make a paragraph for third answer option:

Make an input of type radio:

name attribute is “twoAnswer”

id attribute is “C”

value attribute is also “C”

label attribute is “Disagree”

Make a paragraph for fourth answer option:

Make an input of type radio:

name attribute is “twoAnswer”

id attribute is “D”

value attribute is also “D”

label attribute is “Strongly Disagree”

Make a paragraph for submit button:

Button type is “submit” and the text will be “next”

Make a form for back button:

Button type is “button”, onclick event is “history.back()” method, text is “Back”

Q3 class

Have a chunk of python code:

Import request library

Request the “twoAnswer” from the previous page and set it equal to “twoAnswer”

Request the “score” variable from the previous page and set it equal to “score”

Request the “answers” from the previous page and set it equal to “answers”

Cast score to an int

If twoAnswer is equal to "A":
 Increment score by 20
Otherwise if twoAnswer is equal to "B":
 Increment score by 10
Otherwise if twoAnswer is equal to "C":
 Decrement score by 10
Otherwise if twoAnswer is equal to "D":
 Decrement score by 20
Otherwise, assuming there was no input:
 Set twoAnswer equal to the string 0
End the python block

Concatenate "twoAnswer" and a space key to answers variable
End the python code

Set up the doctype
Declare the html to be english
In the head container:
 Make the title of the tab "Question3"
 Make the charset UTF-8
 add a link to quizStyle.css, rel "stylesheet" and type attribute "text/css"
 have a reference link to the google fonts for "Cormorant Upright" and outline effect and relative attribute "stylesheet"

In the body container:
 Make a level 1 header saying "Question 3"
 Make a paragraph asking the Cholerick question "In a peer group, you are inclined to lead the team"
 Make a form with action attribute that goes to "/q4" and post method attribute:
 Make an input of type "hidden" that will send the data without the user seeing it:
 Name attribute will be "score"
 Value attribute will be the python value of "score"
 Make an input of type "hidden" that will send the data without the user seeing it:
 Name attribute will be "answers"
 Value attribute will be the python value of "answers"
 Make a paragraph for first answer option:
 Make an input of type radio:
 name attribute is "threeAnswer"
 id attribute is "A"
 value attribute is also "A"
 label attribute is "Strongly Agree"
 Make a paragraph for second answer option:
 Make an input of type radio:
 name attribute is "threeAnswer"
 id attribute is "B"

value attribute is also "B"

label attribute is "Agree"

Make a paragraph for third answer option:

Make an input of type radio:

name attribute is "threeAnswer"

id attribute is "C"

value attribute is also "C"

label attribute is "Disagree"

Make a paragraph for fourth answer option:

Make an input of type radio:

name attribute is "threeAnswer"

id attribute is "D"

value attribute is also "D"

label attribute is "Strongly Disagree"

Make a paragraph for submit button:

Button type is "submit" and the text will be "next"

Make a form for back button:

Button type is "button", onclick event is "history.back()" method, text is "Back"

|

Q4 class

|

Have a chunk of python code:

Import request library

Request the "threeAnswer" from the previous page and set it equal to "threeAnswer"

Request the "score" variable from the previous page and set it equal to "score"

Request the "answers" from the previous page and set it equal to "answers"

Cast score to an int

If threeAnswer is equal to "A":

Increment score by 200

Otherwise if threeAnswer is equal to "B":

Increment score by 100

Otherwise if threeAnswer is equal to "C":

Decrement score by 100

Otherwise if threeAnswer is equal to "D":

Decrement score by 200

Otherwise, assuming there was no input:

Set threeAnswer equal to the string 0

End the python block

Concatenate "threeAnswer" and a space key to answers variable

End the python code

Set up the doctype

Declare the html to be english

In the head container:

Make the title of the tab "Question4"

Make the charset UTF-8

add a link to quizStyle.css, rel "stylesheet" and type attribute "text/css"

have a reference link to the google fonts for "Cormorant Upright" and outline effect and relative attribute "stylesheet"

In the body container:

Make a level 1 header saying "Question 4"

Make a paragraph asking the Phlegmatic question "You are happy to let others make decisions for the group"

Make a form with action attribute that goes to "/q5" and post method attribute:

Make an input of type "hidden" that will send the data without the user seeing it:

Name attribute will be "score"

Value attribute will be the python value of "score"

Make an input of type "hidden" that will send the data without the user seeing it:

Name attribute will be "answers"

Value attribute will be the python value of "answers"

Make a paragraph for first answer option:

Make an input of type radio:

name attribute is "fourAnswer"

id attribute is "A"

value attribute is also "A"

label attribute is "Strongly Agree"

Make a paragraph for second answer option:

Make an input of type radio:

name attribute is "fourAnswer"

id attribute is "B"

value attribute is also "B"

label attribute is "Agree"

Make a paragraph for third answer option:

Make an input of type radio:

name attribute is "fourAnswer"

id attribute is "C"

value attribute is also "C"

label attribute is "Disagree"

Make a paragraph for fourth answer option:

Make an input of type radio:

name attribute is "fourAnswer"

id attribute is "D"

value attribute is also "D"

label attribute is "Strongly Disagree"

Make a paragraph for submit button:

Button type is "submit" and the text will be "next"

Make a form for back button:

Button type is "button", onclick event is "history.back()" method, text is "Back"

Q5 class

Have a chunk of python code:

Import request library

Request the "fourAnswer" from the previous page and set it equal to "fourAnswer"

Request the "score" variable from the previous page and set it equal to "score"

Request the "answers" from the previous page and set it equal to "answers"

Cast score to an int

If fourAnswer is equal to "A":

Increment score by 2

Otherwise if fourAnswer is equal to "B":

Increment score by 1

Otherwise if fourAnswer is equal to "C":

Decrement score by 1

Otherwise if fourAnswer is equal to "D":

Decrement score by 2

Otherwise, assuming there was no input:

Set fourAnswer equal to the string 0

End the python block

Concatenate "fourAnswer" and a space key to answers variable

End the python code

Set up the doctype

Declare the html to be english

In the head container:

Make the title of the tab "Question5"

Make the charset UTF-8

add a link to quizStyle.css, rel "stylesheet" and type attribute "text/css"

have a reference link to the google fonts for "Cormorant Upright" and outline effect and relative attribute "stylesheet"

In the body container:

Make a level 1 header saying "Question 5"

Make a paragraph asking the Melancholic question "You are an overthinker"

Make a form with action attribute that goes to "/q6" and post method attribute:

Make an input of type "hidden" that will send the data without the user seeing it:

Name attribute will be “score”

Value attribute will be the python value of “score”

Make an input of type “hidden” that will send the data without the user seeing it:

Name attribute will be “answers”

Value attribute will be the python value of “answers”

Make a paragraph for first answer option:

Make an input of type radio:

name attribute is “fiveAnswer”

id attribute is “A”

value attribute is also “A”

label attribute is “Strongly Agree”

Make a paragraph for second answer option:

Make an input of type radio:

name attribute is “fiveAnswer”

id attribute is “B”

value attribute is also “B”

label attribute is “Agree”

Make a paragraph for third answer option:

Make an input of type radio:

name attribute is “fiveAnswer”

id attribute is “C”

value attribute is also “C”

label attribute is “Disagree”

Make a paragraph for fourth answer option:

Make an input of type radio:

name attribute is “fiveAnswer”

id attribute is “D”

value attribute is also “D”

label attribute is “Strongly Disagree”

Make a paragraph for submit button:

Button type is “submit” and the text will be “next”

Make a form for back button:

Button type is “button”, onclick event is “history.back()” method, text is “Back”

Q6 class

Have a chunk of python code:

Import request library

Request the “fiveAnswer” from the previous page and set it equal to “fiveAnswer”

Request the “score” variable from the previous page and set it equal to “score”

Request the “answers” from the previous page and set it equal to “answers”

Cast score to an int

If fiveAnswer is equal to "A":
 Increment score by 20
Otherwise if fiveAnswer is equal to "B":
 Increment score by 10
Otherwise if fiveAnswer is equal to "C":
 Decrement score by 10
Otherwise if fiveAnswer is equal to "D":
 Decrement score by 20
Otherwise, assuming there was no input:
 Set fiveAnswer equal to the string 0
End the python block

Concatenate "fiveAnswer" and a space key to answers variable

Dealing with possible negatives if two "Strongly Disagrees" for Melancholic were chosen:
if the letter at the second index and the eighth index of answers are both "D":
 add 20 back to score, since it was decremented in the previous calculation
 take the string value of score modulus 10, set it equal to local variable "phl"
 set local variable "mel" to the string 0
 take the string value of the floor division of score/100 modulus 10, set it equal to "cho" variable
 take the string value of the floor division of score/1000 set it equal to "sag" variable
 make score the integer value of sag, cho, mel and phl concatenated together

End the python code

Set up the doctype

Declare the html to be english

In the head container:

 Make the title of the tab "Question6"

 Make the charset UTF-8

 add a link to quizStyle.css, rel "stylesheet" and type attribute "text/css"

 have a reference link to the google fonts for "Cormorant Upright" and outline effect and relative attribute "stylesheet"

In the body container:

 Make a level 1 header saying "Question 6"

 Make a paragraph asking the Choleric question "You are unwilling to give up an opportunity you came across to someone who may need it more"

 Make a form with action attribute that goes to "/q7" and post method attribute:

 Make an input of type "hidden" that will send the data without the user seeing it:

 Name attribute will be "score"

 Value attribute will be the python value of "score"

 Make an input of type "hidden" that will send the data without the user seeing it:

Name attribute will be “answers”

Value attribute will be the python value of “answers”

Make a paragraph for first answer option:

Make an input of type radio:

name attribute is “sixAnswer”

id attribute is “A”

value attribute is also “A”

label attribute is “Strongly Agree”

Make a paragraph for second answer option:

Make an input of type radio:

name attribute is “sixAnswer”

id attribute is “B”

value attribute is also “B”

label attribute is “Agree”

Make a paragraph for third answer option:

Make an input of type radio:

name attribute is “sixAnswer”

id attribute is “C”

value attribute is also “C”

label attribute is “Disagree”

Make a paragraph for fourth answer option:

Make an input of type radio:

name attribute is “sixAnswer”

id attribute is “D”

value attribute is also “D”

label attribute is “Strongly Disagree”

Make a paragraph for submit button:

Button type is “submit” and the text will be “next”

Make a form for back button:

Button type is “button”, onclick event is “history.back()” method, text is “Back”

|

Q7 class

|

Have a chunk of python code:

Import request library

Request the “sixAnswer” from the previous page and set it equal to “sixAnswer”

Request the “score” variable from the previous page and set it equal to “score”

Request the “answers” from the previous page and set it equal to “answers”

Cast score to an int

Dealing with possible negatives if two “Strongly Disagrees” for Choleric were chosen:

If the floor division of score divided by 100 modulus 10 is 1 and score is negative:

Make the floor division of score divided by 100 modulus 10 equal to 0

Take the absolute value of score and set it equal to score

If sixAnswer is equal to "A":
 Increment score by 200
Otherwise if sixAnswer is equal to "B":
 Increment score by 100
Otherwise if sixAnswer is equal to "C":
 Decrement score by 100
Otherwise if sixAnswer is equal to "D":
 Decrement score by 2000
Otherwise, assuming there was no input:
 Set sixAnswer equal to the string 0
End the python block

Concatenate "sixAnswer" and a space key to answers variable

Dealing with possible negatives/score update errors if two "Strongly Disagrees" for Choleric were chosen:

if the letter at the fourth index and the tenth index of answers are both "D":
 add 200 back to score, since it was decremented in the previous calculation
 take the string value of score modulus 10, set it equal to local variable "phl"
 take the string value of the floor division of score/10 modulus 10, set it equal to "mel" variable
 set local variable "cho" to the string 0
 take the string value of the floor division of score/1000 set it equal to "sag" variable
 make score the integer value of sag, cho, mel and phl concatenated together

End the python code

Set up the doctype

Declare the html to be english

In the head container:

 Make the title of the tab "Question7"

 Make the charset UTF-8

 add a link to quizStyle.css, rel "stylesheet" and type attribute "text/css"

 have a reference link to the google fonts for "Cormorant Upright" and outline effect and relative attribute "stylesheet"

In the body container:

 Make a level 1 header saying "Question 7"

 Make a paragraph asking the Saguine question "Someone you know asks you if you want to road trip to the state over for a YouTube video. You are likely to join them"

 Make a form with action attribute that goes to "/q8" and post method attribute:

 Make an input of type "hidden" that will send the data without the user seeing it:

 Name attribute will be "score"

 Value attribute will be the python value of "score"

 Make an input of type "hidden" that will send the data without the user seeing it:

Name attribute will be “answers”

Value attribute will be the python value of “answers”

Make a paragraph for first answer option:

Make an input of type radio:

name attribute is “sevenAnswer”

id attribute is “A”

value attribute is also “A”

label attribute is “Strongly Agree”

Make a paragraph for second answer option:

Make an input of type radio:

name attribute is “sevenAnswer”

id attribute is “B”

value attribute is also “B”

label attribute is “Agree”

Make a paragraph for third answer option:

Make an input of type radio:

name attribute is “sevenAnswer”

id attribute is “C”

value attribute is also “C”

label attribute is “Disagree”

Make a paragraph for fourth answer option:

Make an input of type radio:

name attribute is “sevenAnswer”

id attribute is “D”

value attribute is also “D”

label attribute is “Strongly Disagree”

Make a paragraph for submit button:

Button type is “submit” and the text will be “next”

Make a form for back button:

Button type is “button”, onclick event is “history.back()” method, text is “Back”

|

Q8 class

|

Have a chunk of python code:

Import request library

Request the “sevenAnswer” from the previous page and set it equal to “sevenAnswer”

Request the “score” variable from the previous page and set it equal to “score”

Request the “answers” from the previous page and set it equal to “answers”

Cast score to an int

Dealing with possible negatives if two “Strongly Disagree” for Saguine were chosen:

If the floor division of score divided by 1000 is 1 and score is negative:

Make the floor division of score divided by 1000 equal to 0

Take the absolute value of score and set it equal to score

If sevenAnswer is equal to "A":
 Increment score by 2000
Otherwise if sevenAnswer is equal to "B":
 Increment score by 1000
Otherwise if sevenAnswer is equal to "C":
 Decrement score by 1000
Otherwise if sevenAnswer is equal to "D":
 Decrement score by 2000
Otherwise, assuming there was no input:
 Set sevenAnswer equal to the string 0
End the python block

Concatenate "sevenAnswer" and a space key to answers variable

Dealing with possible negatives/score update errors if two "Strongly Disagrees" for Saguine were chosen:

if the letter at the zeroth index and the twelfth index of answers are both "D":
 add 2000 back to score, since it was decremented in the previous calculation
 take the string value of score modulus 10, set it equal to local variable "phl"
 take the string value of the floor division of score/10 modulus 10, set it equal to "mel" variable
 take the string value of the floor division of score/100 modulus 10, set it equal to "cho" variable
 set local variable "sag" to the string value of 0
 make score the integer value of sag, cho, mel and phl concatenated together

End the python code

Set up the doctype

Declare the html to be english

In the head container:

 Make the title of the tab "Question8"

 Make the charset UTF-8

 add a link to quizStyle.css, rel "stylesheet" and type attribute "text/css"

 have a reference link to the google fonts for "Cormorant Upright" and outline effect and relative attribute "stylesheet"

In the body container:

 Make a level 1 header saying "Question 8"

 Make a paragraph asking the Phlegmatic question "You can be relied on to get the job done"

 Make a form with action attribute that goes to "/q9" and post method attribute:

 Make an input of type "hidden" that will send the data without the user seeing it:

 Name attribute will be "score"

 Value attribute will be the python value of "score"

 Make an input of type "hidden" that will send the data without the user seeing it:

 Name attribute will be "answers"

Value attribute will be the python value of “answers”

Make a paragraph for first answer option:

Make an input of type radio:

name attribute is “eightAnswer”

id attribute is “A”

value attribute is also “A”

label attribute is “Strongly Agree”

Make a paragraph for second answer option:

Make an input of type radio:

name attribute is “eightAnswer”

id attribute is “B”

value attribute is also “B”

label attribute is “Agree”

Make a paragraph for third answer option:

Make an input of type radio:

name attribute is “eightAnswer”

id attribute is “C”

value attribute is also “C”

label attribute is “Disagree”

Make a paragraph for fourth answer option:

Make an input of type radio:

name attribute is “eightAnswer”

id attribute is “D”

value attribute is also “D”

label attribute is “Strongly Disagree”

Make a paragraph for submit button:

Button type is “submit” and the text will be “next”

Make a form for back button:

Button type is “button”, onclick event is “history.back()” method, text is “Back”

|

Q9 class

|

Have a chunk of python code:

Import request library

Request the “eightAnswer” from the previous page and set it equal to “eightAnswer”

Request the “score” variable from the previous page and set it equal to “score”

Request the “answers” from the previous page and set it equal to “answers”

If the previous score ends in 0.1 because of phlegmatic negatives:

convert score into a float

otherwise:

cast score into a float before casting it into an int

Dealing with possible negatives if two “Strongly Disagrees” for Phlegmatic were chosen:

If the floor division of score modulus 10 is 1 and score is negative:

Make the floor division of score modulus 10 equal to 0
Take the absolute value of score and set it equal to score
If eightAnswer is equal to "A":
 Increment score by 2
Otherwise if eightAnswer is equal to "B":
 Increment score by 1
Otherwise if eightAnswer is equal to "C":
 Decrement score by 1
Otherwise if eightAnswer is equal to "D":
 Decrement score by 2
Otherwise, assuming there was no input:
 Set eightAnswer equal to the string 0
End the python block

Concatenate "eightAnswer" and a space key to answers variable

Dealing with possible negatives/score update errors if two "Strongly Disagrees" for Phlegmatic were chosen:

if the letter at the sixth index and the fourteenth index of answers are both "D":
 add 2 back to score, since it was decremented in the previous calculation
 set local variable "phl" to the string value of 0.1 to not ruin digit values
 take the string value of the floor division of score/10 modulus 10, set it equal to "mel" variable
 take the string value of the floor division of score/100 modulus 10, set it equal to "cho" variable
 take the string value of the floor division of score/1000 set it equal to "sag" variable
 make score the integer value of sag, cho, mel and phl concatenated together

End the python code

Set up the doctype

Declare the html to be english

In the head container:

 Make the title of the tab "Question9"

 Make the charset UTF-8

 add a link to quizStyle.css, rel "stylesheet" and type attribute "text/css"

 have a reference link to the google fonts for "Cormorant Upright" and outline effect and relative attribute "stylesheet"

In the body container:

 Make a level 1 header saying "Question 9"

 Make a paragraph asking the Choleric question "You are assertive and confident"

 Make a form with action attribute that goes to "/q10" and post method attribute:

 Make an input of type "hidden" that will send the data without the user seeing it:

 Name attribute will be "score"

 Value attribute will be the python value of "score"

Make an input of type “hidden” that will send the data without the user seeing it:

Name attribute will be “answers”

Value attribute will be the python value of “answers”

Make a paragraph for first answer option:

Make an input of type radio:

name attribute is “nineAnswer”

id attribute is “A”

value attribute is also “A”

label attribute is “Strongly Agree”

Make a paragraph for second answer option:

Make an input of type radio:

name attribute is “nineAnswer”

id attribute is “B”

value attribute is also “B”

label attribute is “Agree”

Make a paragraph for third answer option:

Make an input of type radio:

name attribute is “nineAnswer”

id attribute is “C”

value attribute is also “C”

label attribute is “Disagree”

Make a paragraph for fourth answer option:

Make an input of type radio:

name attribute is “nineAnswer”

id attribute is “D”

value attribute is also “D”

label attribute is “Strongly Disagree”

Make a paragraph for submit button:

Button type is “submit” and the text will be “next”

Make a form for back button:

Button type is “button”, onclick event is “history.back()” method, text is “Back”

Q10 class

Have a chunk of python code:

Import request library

Request the “nineAnswer” from the previous page and set it equal to “nineAnswer”

Request the “score” variable from the previous page and set it equal to “score”

Request the “answers” from the previous page and set it equal to “answers”

If the previous score ends in 0.1 because of phlegmatic negatives:

convert score into a float

otherwise:

```
    cast score into a float before casting it into an int
End the python block
If the floor division of score/100 modulus 10 is greater than 0:
    If nineAnswer is equal to "A":
        Increment score by 200
    Otherwise if nineAnswer is equal to "B":
        Increment score by 100
    Otherwise if nineAnswer is equal to "C":
        Decrement score by 100
    Otherwise if nineAnswer is equal to "D":
        if at least one of the previous two choleric qs were not D:
            If the floor division of score/100 modulus 10 is greater than 1:
                Decrement score by 200
            Otherwise:
                Decrement score by 100
    End the python block
Otherwise, assuming there was no input:
    Set nineAnswer equal to the string 0
End the python block
```

```
Concatenate "nineAnswer" and a space key to answers variable
End the python code
```

```
Set up the doctype
Declare the html to be english
In the head container:
    Make the title of the tab "Question10"
    Make the charset UTF-8
    add a link to quizStyle.css, rel "stylesheet" and type attribute "text/css"
    have a reference link to the google fonts for "Cormorant Upright" and outline effect and relative attribute
    "stylesheet"
```

```
In the body container:
    Make a level 1 header saying "Question 10"
    Make a paragraph asking the Melancholic question "You are logical person who finds it difficult to open
up"
    Make a form with action attribute that goes to "/q11" and post method attribute:
        Make an input of type "hidden" that will send the data without the user seeing it:
            Name attribute will be "score"
            Value attribute will be the python value of "score"
        Make an input of type "hidden" that will send the data without the user seeing it:
            Name attribute will be "answers"
            Value attribute will be the python value of "answers"
    Make a paragraph for first answer option:
```

Make an input of type radio:

name attribute is "tenAnswer"

id attribute is "A"

value attribute is also "A"

label attribute is "Strongly Agree"

Make a paragraph for second answer option:

Make an input of type radio:

name attribute is "tenAnswer"

id attribute is "B"

value attribute is also "B"

label attribute is "Agree"

Make a paragraph for third answer option:

Make an input of type radio:

name attribute is "tenAnswer"

id attribute is "C"

value attribute is also "C"

label attribute is "Disagree"

Make a paragraph for fourth answer option:

Make an input of type radio:

name attribute is "tenAnswer"

id attribute is "D"

value attribute is also "D"

label attribute is "Strongly Disagree"

Make a paragraph for submit button:

Button type is "submit" and the text will be "next"

Make a form for back button:

Button type is "button", onclick event is "history.back()" method, text is "Back"

Q11 class

Have a chunk of python code:

Import request library

Request the "tenAnswer" from the previous page and set it equal to "tenAnswer"

Request the "score" variable from the previous page and set it equal to "score"

Request the "answers" from the previous page and set it equal to "answers"

If the previous score ends in 0.1 because of phlegmatic negatives:

convert score into a float

otherwise:

cast score into a float before casting it into an int

End the python block

If the floor division of score/10 modulus 10 is greater than 0:

If tenAnswer is equal to "A":

Increment score by 20

Otherwise if tenAnswer is equal to "B":
 Increment score by 10
Otherwise if tenAnswer is equal to "C":
 Decrement score by 10
Otherwise if tenAnswer is equal to "D":
 If the floor division of score/10 modulus 10 is greater than 1:
 Decrement score by 20
 Otherwise:
 Decrement score by 10
 End the python block
Otherwise, assuming there was no input:
 Set tenAnswer equal to the string 0
End the python block

Concatenate "tenAnswer" and a space key to answers variable
End the python code

Set up the doctype
Declare the html to be english
In the head container:
 Make the title of the tab "Question11"
 Make the charset UTF-8
 add a link to quizStyle.css, rel "stylesheet" and type attribute "text/css"
 have a reference link to the google fonts for "Cormorant Upright" and outline effect and relative attribute "stylesheet"

In the body container:
 Make a level 1 header saying "Question 11"
 Make a paragraph asking the Saguine question "You can be referred to as the life of the party"
 Make a form with action attribute that goes to "/q12" and post method attribute:
 Make an input of type "hidden" that will send the data without the user seeing it:
 Name attribute will be "score"
 Value attribute will be the python value of "score"
 Make an input of type "hidden" that will send the data without the user seeing it:
 Name attribute will be "answers"
 Value attribute will be the python value of "answers"
 Make a paragraph for first answer option:
 Make an input of type radio:
 name attribute is "elevenAnswer"
 id attribute is "A"
 value attribute is also "A"
 label attribute is "Strongly Agree"
 Make a paragraph for second answer option:
 Make an input of type radio:

name attribute is "elevenAnswer"

id attribute is "B"

value attribute is also "B"

label attribute is "Agree"

Make a paragraph for third answer option:

Make an input of type radio:

name attribute is "elevenAnswer"

id attribute is "C"

value attribute is also "C"

label attribute is "Disagree"

Make a paragraph for fourth answer option:

Make an input of type radio:

name attribute is "elevenAnswer"

id attribute is "D"

value attribute is also "D"

label attribute is "Strongly Disagree"

Make a paragraph for submit button:

Button type is "submit" and the text will be "next"

Make a form for back button:

Button type is "button", onclick event is "history.back()" method, text is "Back"

Q12 class

Have a chunk of python code:

Import request library

Request the "elevenAnswer" from the previous page and set it equal to "elevenAnswer"

Request the "score" variable from the previous page and set it equal to "score"

Request the "answers" from the previous page and set it equal to "answers"

If the previous score ends in 0.1 because of phlegmatic negatives:

convert score into a float

otherwise:

cast score into a float before casting it into an int

End the python block

If the floor division of score/1000 is greater than 0:

If elevenAnswer is equal to "A":

Increment score by 2000

Otherwise if elevenAnswer is equal to "B":

Increment score by 1000

Otherwise if elevenAnswer is equal to "C":

Decrement score by 1000

Otherwise if elevenAnswer is equal to "D":

If the floor division of score/1000 is greater than 1:

Decrement score by 2000

Otherwise:
Decrement score by 1000
End the python block
Otherwise, assuming there was no input:
Set elevenAnswer equal to the string 0
End the python block

Concatenate “elevenAnswer” and a space key to answers variable
End the python code

Set up the doctype
Declare the html to be english
In the head container:
Make the title of the tab “Question12”
Make the charset UTF-8
add a link to quizStyle.css, rel "stylesheet" and type attribute "text/css"
have a reference link to the google fonts for "Cormorant Upright" and outline effect and relative attribute "stylesheet"

In the body container:
Make a level 1 header saying “Question 12”
Make a paragraph asking the Phlegmatic question “You are known to give good advice, but you yourself are unambitious”
Make a form with action attribute that goes to “/qResult” and post method attribute:
Make an input of type “hidden” that will send the data without the user seeing it:
Name attribute will be “score”
Value attribute will be the python value of “score”
Make an input of type “hidden” that will send the data without the user seeing it:
Name attribute will be “answers”
Value attribute will be the python value of “answers”
Make a paragraph for first answer option:
Make an input of type radio:
name attribute is “twelveAnswer”
id attribute is “A”
value attribute is also “A”
label attribute is “Strongly Agree”
Make a paragraph for second answer option:
Make an input of type radio:
name attribute is “twelveAnswer”
id attribute is “B”
value attribute is also “B”
label attribute is “Agree”
Make a paragraph for third answer option:
Make an input of type radio:

name attribute is "twelveAnswer"

id attribute is "C"

value attribute is also "C"

label attribute is "Disagree"

Make a paragraph for fourth answer option:

Make an input of type radio:

name attribute is "twelveAnswer"

id attribute is "D"

value attribute is also "D"

label attribute is "Strongly Disagree"

Make a paragraph for submit button:

Button type is "submit" and the text will be "next"

Make a form for back button:

Button type is "button", onclick event is "history.back()" method, text is "Back"

qResult Class

Have a chunk of python code:

Import request library

Request the "twelveAnswer" from the previous page and set it equal to "twelveAnswer"

Request the "score" variable from the previous page and set it equal to "score"

Request the "answers" from the previous page and set it equal to "answers"

If the previous score ends in 0.1 because of phlegmatic negatives:

convert score into a float

otherwise:

cast score into a float before casting it into an int

end the python block

If the integer value of score modulus 10 is greater than 0:

If twelveAnswer is equal to "A":

Increment score by 2

Otherwise if twelveAnswer is equal to "B":

Increment score by 1

Otherwise if twelveAnswer is equal to "C":

Decrement score by 1

Otherwise if twelveAnswer is equal to "D":

If the integer value of score modulus 10 is greater than 1:

Decrement score by 2

Otherwise:

Decrement score by 1

End the python block

Otherwise, assuming there was no input:

Set twelveAnswer equal to the string 0

End the python block
End the python block

Concatenate “twelveAnswer” and a space key to answers variable
Split the answers string based on the space delimiter, cast it into a list, and set it equal to “answers”

By this point, score should theoretically be a 4 digit number
Store the integer result of score modulus 10 into a local ones variable
Do a floor division of score divided by 10
Store the result of score modulus 10 into a local tens variable
Do a floor division of score divided by 10
Store the result of score modulus 10 into a local hundreds variable
Do a floor division of score divided by 10
Store the remaining score variable into a local thousands variable

Largest variable will be created and defaulted to the string thousands variable value (since Saguine is the most common personality temperament)

If the hundreds value is greater than the largest value:

 largest is set equal to the string hundreds value

 largestValue is set equal to the hundreds value

end the python block

If the tens value is greater than largestValue:

 largest is set equal to the string tens value

 largestValue is set equal to the tens value

end the python block

If the ones value is greater than largestValue:

 largest is set equal to the string ones value

 largestValue is set equal to the ones value

end the python block

Handling ties:

If largest is not the thousands variable (Saguine wins all ties by default):

 Make local int pStrong, assign it to a value of 0

 Make local int mStrong, assign it to a value of 0

 Make local int cStrong, assign it to a value of 0

 Have a for loop with variable “i” and a range the length of the answers list:

 Make local variable qNum, assign it to i+1

 If qNum is divisible by 4 (on a phlegmatic question):

 If the answers value at position “i” is “A”:

 Increment pStrong

 End the python block

 End the python block

 If qNum is divisible by 5 or qNum is 2 (on a melancholic question):

 If the answers value at position “i” is “A”:

```

    Increment mStrong
End the python block
End the python block
If qNum is divisible by 3 and not 12 (on a choleric question):
    If the answers value at position "i" is "A":
        Increment cStrong
    End the python block
End the python block
End the python block
If largest is the tens variable (can't be ones if tie):
    If the tens value is equal to the ones value (M/P tie):
        If pStrong is greater than or equal to mStrong:
            Set largest equal to string ones
            Set largestValue equal to ones
        Otherwise if largest is the hundreds variable:
            If the hundreds value is equal to the ones value (C/P tie):
                If cStrong is less than pStrong:
                    Set largest equal to string ones
                    Set largestValue equal to ones
                Otherwise if the hundreds value is equal to the tens value:
                    If cStrong is less than mStrong:
                        Set largest equal to string tens
                        Set largestValue equal to tens
                    End the python block
                End the python block
            End the python block
        End the python block
    End the python block
end the python code

Set up the doctype
Declare the html to be english
In the head container:
    Make the title of the tab "CalculatingResult"
    Make the charset UTF-8
    add a link to quizStyle.css, rel "stylesheet" and type attribute "text/css"
    have a reference link to the google fonts for "Cormorant Upright" and outline effect and relative attribute
    "stylesheet"

In the body container:
    Have a level one header saying "Almost there..."
    Make a form with action attribute "{ { largest } }" and post method attribute:
        Make a paragraph saying "To confirm your answers, press the button. To go back, press the back
        button."
        Make a button of type submit with "Submit Answers" written on it

```

Make a form for back button:

Button type is "button", onclick event is "history.back()" method, text is "Back"

|

Ones class

|

The user will be sent here if they have been categorized into Phlegmatic

Set up the doctype

Declare the html to be english

In the head container:

Make the title of the tab "Phlegmatic Result"

Make the charset UTF-8

In the body container:

Make a level one header with text "You got: Phlegmatic!"

Make a paragraph explaining what phlegmatic means.

Make a paragraph with a list of three strengths

Make a paragraph with a list of three weaknesses

Make a form with action attribute /start and post method attribute:

Make a button of type "submit" with name "Take Quiz Again"

|

Tens class

|

The user will be sent here if they have been categorized into Melancholic

Set up the doctype

Declare the html to be english

In the head container:

Make the title of the tab "Melancholic"

Make the charset UTF-8

In the body container:

Make a level one header with text "You got: Melancholic!"

Make a paragraph explaining what melancholic means.

Make a paragraph with a list of three strengths

Make a paragraph with a list of three weaknesses

Make a form with action attribute /start and post method attribute:

Make a button of type "submit" with name "Take Quiz Again"

|

Hundreds class

|

The user will be sent here if they have been categorized into Choleric

Set up the doctype

Declare the html to be english

In the head container:

Make the title of the tab “Choleric”

Make the charset UTF-8

In the body container:

Make a level one header with text “You got: Choleric!”

Make a paragraph explaining what choleric means.

Make a paragraph with a list of three strengths

Make a paragraph with a list of three weaknesses

Make a form with action attribute /start and post method attribute:

Make a button of type “submit” with name “Take Quiz Again”

|

Thousands class

|

The user will be sent here if they have been categorized into Saguine

Set up the doctype

Declare the html to be english

In the head container:

Make the title of the tab “Saguine”

Make the charset UTF-8

In the body container:

Make a level one header with text “You got: Saguine!”

Make a paragraph explaining what saguine means.

Make a paragraph with a list of three strengths

Make a paragraph with a list of three weaknesses

Make a form with action attribute /start and post method attribute:

Make a button of type “submit” with name “Take Quiz Again”

|

Works Cited:

Pages used to research temperaments:

- [What are the 16 Personality Types? By \[Myer Briggs\] + Free Tests \(formpl.us\)](#)
- [Which Type Are You? The Four Personality Types Explained | Slism](#)
- [four temperament personality types - Bing](#)
- [four temperament personality types - Bing](#)

Score variable logic:

- [Create a personality quiz - Help Center | Typeform](#)

Python help:

- [7 Powerful ways to Convert string to list in Python - Python Pool](#)
- Canvas announcement about end statements

HTML help:

- [html - How to send hidden data - Stack Overflow](#)
- [How to create an HTML back button \(computerhope.com\)](#)

CSS:

- [CSS Fonts \(w3schools.com\)](#)

Link to project: [Start](#)