

Let's Code - Test

Die "Let's Code" Lernplattform soll explorativ getestet werden. Das heißt, dass der Tester im Laufe des Tests alle Funktionen der Webseite durchgeht und eventuelle Fehler meldet. Da auch einige komplexere Testcases vorkommen, werden diese hier nochmal detailliert mit den einzelnen Schritten und dem erwarteten Ergebnis beschrieben. Hier sollen die Ergebnisse direkt in der Tabelle eingetragen werden, sodass direkt verglichen werden kann, ob das erwartete Ergebnis mit dem eigentlichen Ergebnis übereinstimmt.

Webseite

Kommen wir zuerst zum Gesamteindruck der Seite:

Layout	Wie gefällt dir das Layout der Startseite/des Intros?	
Designentscheidungen	Ist der Aufbau der Webseite logisch? Findet man leicht zur gewünschten Funktion(Intro der C# Lernplattform, zurück zur Startseite)?	

Lernplattform

Als nächstes sollen die einzelnen Level getestet werden. In jedem Level wird ein neues Element der Programmiersprache C# erklärt und dann in einer oder mehreren Aufgaben auch praktisch geübt. Ab Level 7 arbeitest du dann an einem Projekt, welches am Ende ein IDLE Game ergibt. Die vollständige Referenzlösung ist in Level 14 zu finden.

Beim Test sollen alle Level durchgegangen und auch durchgeführt werden. Wenn dabei Fehler auftreten oder etwas Unerwartetes passiert, bitte am Ende der Seite bei den Notizen notieren.

Scenario	Test Steps	Input Data	Expected Result	Actual Result
1. Überprüfe die Funktionalität der Buttons	1. Klick Button "Learn C#" 2. Klick Button "Startseite" 3. Klick Button "Learn C#" 4. Teste die "Levelnavigation" 5. Klick Button "run"		1. Aufruf des Intro 2. Aufruf der Startseite 3. Aufruf des Intro 4. Aufruf des korrespondierenden Levels 5. Validierung des Codes	
2. Kontrolliere, ob unter dem Punkt "Referenzlösung" Download von Daten möglich ist	1. Navigiere zum Tab der Referenzlösung 2. Klicke den "Download" Button		Download einer .zip Datei mit Beispielcode und einer Readme mit Anleitung zum Öffnen des Codes in Visual Studio	
3. Check Level 1 vordefinierte Code und Validierung	1. Durchlesen der Beschreibung. 2. Kontrolle auf Verständlichkeit und Rechtschreibung 3. Durchführen der Aufgabe 1	Console.WriteLine("Thomas"); ----- string Name = "Thomas"; Console.WriteLine(Name); ----- string Satz = "Ich kann auch ganze Sätze schreiben!"; Console.WriteLine(Satz);	Output: Aufgabe1: PopUp "Aufgabe 1 gelöst!" Aufgabe2: PopUp "Aufgabe " gelöst!" Aufgabe3: PopUp Aufgabe 3 gelöst!	

	4. Durchführen der Aufgabe 2 5. Durchführen der Aufgabe 3 6. Falsches Durchführen der Aufgabe 1 (ohne Strichpunkt) 7. Falsches Durchführen der Aufgabe 1 (beliebig)		Aufgabe1Falsch1: PopUp "Vergiss nicht den Strichpunkt 😊" Aufgabe1Falsch2: PopUp "Leider nicht korrekt!\nDein Output lautet: {output}"	
4. Check Level 4 vordefinierte r Code und Validierung	1. Durchlesen der Beschreibung. 2. Kontrolle auf Verständlichkeit und Rechtschreibung 3. Durchführen der Aufgabe 1 4. Durchführen der Aufgabe 2	Aufgabe 1: int maxHitPoints; int currHitPoints; int currLvl; int attack; int defense; string name; Aufgabe 2: public int maxHitPoints {get;set;} public int currHitPoints{get;set;} public int currLvl{get;set;} public int attack{get;set;} public int defense{get;set;} public string name{get;set;}	Output: Aufgabe1: PopUp "Aufgabe 1 gelöst!" Aufgabe2: PopUp "Aufgabe 2 gelöst!"	
5. Check Level 7 vordefinierte r Code und Validierung	1. Durchlesen der Beschreibung. 2. Kontrolle auf Verständlichkeit und Rechtschreibung 3. Durchführen der Aufgabe	private Monster currentMonster; public Monster CurrentMonster { get { return currentMonster; } set { currentMonster = value; } } private Player player; public Player Player { get { return player; } set { player = value; } } private string message; public string Message { get { return message; } set { message = value; } }	PopUp "Aufgabe 1 gelöst!"	
6. Check Level 8 vordefinierte r Code und Validierung	1. Durchlesen der Beschreibung. 2. Kontrolle auf Verständlichkeit und Rechtschreibung 3. Durchführen der Aufgabe	public int hitpointsBerechnen(int attack, int defence, int dmg) { dmg = (attack - defence); return dmg; }	PopUp "Aufgabe 1 gelöst!"	
7. Check Level 11 vordefinierte r Code und Validierung	1. Durchlesen der Beschreibung. 2. Kontrolle auf Verständlichkeit und Rechtschreibung	for(int i = 0; i < monsters.length; i++){ monsters[i] = new Monster(10 + i, 10 + i, 1+i, 1+i+1, "Monster" + i); }	PopUp "Aufgabe 1 gelöst!"	

	3. Durchführen der Aufgabe			
8. Check Level 12 vordefinierte r Code und Validierung	<ol style="list-style-type: none"> 1. Durchlesen der Beschreibung. 2. Kontrolle auf Verständlichkeit und Rechtschreibung 3. Durchführen der Aufgabe 	<pre>int roundCounter = 0; int hp = 0; switch (roundCounter) { case 1: hp = 1; break; case 2: hp = 2; break; } roundCounter++; return hp;</pre>	PopUp "Aufgabe 1 gelöst!"	
9. Stability check	<ol style="list-style-type: none"> 1. Klicke mehrmals ungezielt durch die Seite 2. Schicke mindestens fünf coding examples zur Evaluierung 3. Lasse die Seite mindestens 5 Minuten lang offen. 	Vordefinierter Code kann verwendet werden	Kein Störverhalten. Selbes Verhalten und Aussehen wie nach frischem Laden der Seite.	

Notizen

zB Auffälligkeiten bei einem Level, Fehler oder generelle Hinweise zu der Webseite, die noch nicht erwähnt wurden