

Assignment 1

nr. 1.

1
2
3

2.

1
2
3

3.

1
2
3

4.

1
2
3

5.

1
2
3

6.

1
2
3

7.

1
2
3

8.

1
2
3

9.

1
2
3

10.

1
2
3

11.

1
2
3

12.

1
2
3

13.

1
2
3

14.

1
2
3

15.

1
2
3

16.

1
2
3

17.

1
2
3

18.

1
2
3

19.

1
2
3

20.

1
2
3

21.

1
2
3

22.

1
2
3

23.

1
2
3

24.

1
2
3

25.

1
2
3

26.

1
2
3

27.

1
2
3

28.

1
2
3

29.

1
2
3

30.

1
2
3

31.

1
2
3

32.

1
2
3

33.

1
2
3

34.

1
2
3

35.

1
2
3

36.

1
2
3

37.

1
2
3

38.

1
2
3

39.

1
2
3

40.

1
2
3

41.

1
2
3

42.

1
2
3

43.

1
2
3

44.

1
2
3

45.

1
2
3

46.

1
2
3

47.

1
2
3

48.

1
2
3

49.

1
2
3

50.

1
2
3

51.

1
2
3

52.

1
2
3

53.

1
2
3

54.

1
2
3

55.

1
2
3

56.

1
2
3

57.

1
2
3

58.

1
2
3

59.

1
2
3

60.

1
2
3

61.

1
2
3

62.

1
2
3

63.

1
2
3

64.

1
2
3

65.

1
2
3

66.

1
2
3

67.

1
2
3

68.

1
2
3

69.

1
2
3

70.

1
2
3

71.

1
2
3

72.

1
2
3

73.

1
2
3

74.

1
2
3

75.

1
2
3

76.

1
2
3

77.

1
2
3

78.

1
2
3

79.

1
2
3

80.

1
2
3

81.

1
2
3

82.

1
2
3

83.

1
2
3

84.

1
2
3

85.

1
2
3

86.

1
2
3

87.

1
2
3

88.

1
2
3

89.

1
2
3

90.

1
2
3

91.

1
2
3

`<button onclick="myfunc"> click me </button>`

JSX
= render()

$\text{window.} \underline{\text{onload}} = \{ \text{ } \}$

to execute some fn on all records
whereage is loaded, we use

forall

The diagrams show a double flower on the left, labeled 'double', and a single flower on the right, labeled 'single'. The double flower has many petals, while the single flower has only five petals. The single flower is also labeled '2nd'.

on markdown

down
up
click

down
up
options menu

DOM, HTML doesn't

mousedown = event.target
mouseup = event.target

< button

ch handles the execution of fn associated with an event, when that event occurs.

```

graph TD
    A[onclick = myfunc()] --> B[Event handler function]
    B --> C[fn]
    C --> D[Click]
    C --> E[mouseover]
    C --> F[mouseout]
    D --> E
    E --> F
  
```

event handler

- 1) add event listener
 - add
 - attach
 - bind
 - listen
 - subscribe
 - on
 - addListener
 - addEventListener
- 2) remove event listener
 - remove
 - detach
 - unbind
 - unlisten
 - unsubscribe
 - off
 - removeListener
 - removeEventListener

[illegible]

1) use interact with widget \rightarrow ch is activated

2) $\rho^{(0)}$ (on)

cup
bubbling

3) event that \leq activities related to student

h) `addEl("child", cb)` → asynchronous JS

1) Event Capturing (TTB)

- ↳ only event handlers will be fired
- ↳ then specific event handler is fired

Event handler \rightarrow capture, target, bubbles

①

DOC x

HTML ^

BODY → click, cb1

DIV → click, cb2

button → click, cb3

③

2) Event bubbling (BTT) -

3) Target

babbings → default
 × ~~light~~ → occur always
 cupbings →

$f_m(a, b, c)$ \nearrow unbekannt \nearrow take