**ПЕРЕМЕННЫЕ**

Let tempMult = fasle //индикатор: «сейчас умножение»

Let tempMultPlus = fasle //индикатор: «сейчас приоритетное умножение перед плюсом»

Let tempMultMinus = fasle //индикатор: «сейчас приоритетное умножение перед минусом»

Let saveMult //для временного сохранения предыдущего для «при умножении»

Let saveMultPlus //для временного сохранения предыдущего для «при приоритетном умножении перед плюсом»

Let saveMultMinus //для временного сохранения предыдущего для «при приоритетном умножении перед минусом»

Let tempSplit = fasle //индикатор: «сейчас деление»

Let tempSplitPlus = fasle //индикатор: «сейчас приоритетное деление перед плюсом»

Let tempSplitMinus = fasle //индикатор: «сейчас приоритетное деление перед минусом»

Let saveSplit //для временного сохранения предыдущего для при делении

Let saveSplitPlus //для временного сохранения предыдущего для «при приоритетном делении перед плюсом»

Let saveSplitMinus //для временного сохранения предыдущего для при приоритетном делении перед минусом»

Let tempMinus = fasle //индикатор: «сейчас минус»

Let tempPlus = fasle //индикатор: «сейчас плюс»

Let saveMinus //временного сохранения значения индикатора: «сейчас минус»

Let savePlus //временного сохранения значения индикатора: «сейчас плюс»

Let numbers = [] //массив для общего сложения чисел

**КЛАВИШИ ЦИФР**

input0: функция () {

if (сейчас состояние ‘после равно’: (включилось нажатием на ‘=’ после вычислений))

if (tempQ === true) {

1. Вывод равно ‘0’
2. Присвоить темпу значение вывода (‘0’)
3. у вспомогательную переменную темпа занести числовое темп
4. выключить состояние «после равно»
5. title.innerText = ‘0’
6. temp = title.innerText
7. saveNum = parseInt(temp)
8. tempQ = false

} else if (сейчас состояние ‘нуля’: (дефолтное))

else if (title.innerText === ’0’) {

1. Вывод сделать ‘0’
2. Присвоить темпу значение вывода (‘0’)
3. у вспомогательную переменную темпа занести числовое темп
4. title.innerText = 0
5. temp = 0
6. saveNum = parseInt(temp)

} else if (сейчас начальное состояние = '-0': (включилось первым нажатием на -))

else if (title.innerText === ’-0’) {

1. Вывод сделать ‘0’
2. Присвоить темпу значение вывода (‘0’)
3. у вспомогательную переменную темпа занести числовое темп
4. title.innerText = 0
5. temp = 0
6. saveNum = parseInt(temp)

} else (во всех остальных случаях: (набор нулей после цифры) {

1. До вывода добавить ‘0’
2. До темпа добавить ‘0’
3. в вспомогательную переменную темпа занести числовое темп
4. title.innerText += ‘0’
5. temp += ‘0’
6. saveNum = parseInt(temp)

}

},

Input1-9: функция () {

if (сейчас состояние ‘после равно’: (включилось нажатием на ‘=’ после вычислений))

if (tempQ === true) {

1. Очистить вывод и добавить 1-9
2. Присвоить темпу значение вывода (‘1-9’)
3. у вспомогательную переменную темпа занести числовое темп
4. выключить состояние «после равно»
5. title.innerText = (title.innerText = ‘’) + ‘1-9’
6. temp = title.innerText
7. saveNum = parseInt(temp)
8. tempQ = false

} else if (сейчас состояние ‘нуля’: (дефолтное))

else if (title.innerText === ’0’) {

1. Вывод сделать ‘1-9’
2. обнулить темп
3. у вспомогательную переменную темпа занести числовое темп
4. title.innerText = 1-9
5. temp = 0
6. saveNum = parseInt(temp)

} else if (сейчас начальное состояние = '-0': (включилось первым нажатием на -))

else if (title.innerText === ’-0’) {

1. Начальное состояние сделать ‘0’
2. обнулить темп
3. у вспомогательную переменную темпа занести числовое темп
4. title.innerText = 0
5. temp = 0
6. saveNum = parseInt(temp)

} else (во всех остальных случаях: (набор нулей после цифры) {

1. До вывода добавить ‘0’
2. До темпа добавить ‘0’
3. в вспомогательную переменную темпа занести числовое темп
4. title.innerText += ‘0’
5. temp += ‘0’
6. saveNum = parseInt(temp)

}

},

**УМНОЖЕНИЕ**: что делать при нажатии

Умножение: функция () {

// если «не умножение» то проверяем на деление и минус, если нет то потом дефолт «если не умножение»

if (сейчас (было перед этим) умножение: tempMult=true (включилось первым нажатием на \*)

1. Взять последнее с массива в saveMult удаляя его с массива
2. Умножить последнее с массива на сейчас-введенное и занести в массив
3. saveMult = numbers.pop()
4. numbers[numbers.length] = (parseInt(saveMult) \* parseInt(temp))

} else (сейчас (было перед этим) НЕ умножение: tempMult=false (по умолчанию)) {

if (сейчас (было перед этим) деление: tempSplit=true (включилось первым нажатием на /)

1. Взять последнее с массива в saveSplit удаляя с массива
2. Поделить последнее с массива на сейчас-введенное и занести в массив
3. Сбросить состояние деления
4. Сделать состояние умножения
5. saveSplit = numbers.pop()
6. numbers[numbers.length] = (parseInt(saveSplit) / parseInt(temp))
7. tempSplit = false
8. tempMult = true

} else if (сейчас (было перед этим) минус: tempMinus=true (включилось первым нажатием на -)

1. значение в темпе сделать отрицательным и занести в массив
2. Сбросить состояние минуса
3. Сделать глобальное состояние умножения
4. numbers[numbers.length] = (saveNum \* (-1))
5. tempMinus = false
6. tempMult - true

} else {

1. Получить предыдущее значение в массив
2. Сбросить другие состояния (деления, минуса)
3. Сделать состояние «сейчас умножение»: tempMult=true
4. nambers[numbers.length] = saveNum
5. tempMinus = false

tempSplit = false

1. tempMult = true

}

**Деление**: что делать при нажатии

Деление: функция () {

if (сейчас (было перед этим) деление: tempSplit=true (включилось первым нажатием на /){

1. Взять последнее с массива в saveSplit удаляя с массива
2. Поделить последнее с массива на сейчас-введенное и занести в массив
3. saveSplit = numbers.pop()
4. numbers[numbers.length] = (parseInt(saveSplit) / parseInt(temp))

}

else if (сейчас (было перед этим) НЕ деление: tempSplit=false (по умолчанию)) {

if (сейчас (было перед этим) умножение: tempMult=true (включилось первым нажатием на \*)

1. Взять последнее с массива в saveMult удаляя с массива
2. Умножить последнее с массива на сейчас-введенное и занести в массив
3. Сбросить состояние умножения
4. Сделать состояние деления
5. saveMult = numbers.pop()
6. numbers[numbers.length] = (parseInt(saveMult) / parseInt(temp))
7. tempMult = false
8. tempSplit = true

} else if (сейчас (было перед этим) минус: tempMinus=true (включилось первым нажатием на -)

1. значение в темпе сделать отрицательным и занести в массив
2. Сбросить состояние минуса
3. Сделать глобальное состояние умножения
4. numbers[numbers.length] = (saveNum \* (-1))
5. tempMinus = false
6. tempSplit - true

} else {

1. Получить предыдущее значение в массив
2. Сбросить другие состояния (умножения, минуса)
3. Сделать состояние «сейчас деление»: tempSplit=true
4. nambers.[numbers.length] = saveNum
5. tempMinus = false

tempMult = false

1. tempSplit = true

}

**Минус**: что делать при нажатии

Минус: функция () {

if (сейчас (было перед этим) минус: tempMinus=true (включилось первым нажатием на -) {

1. значение в темпе сделать отрицательным и занести в массив
2. numbers[numbers.length] = (saveNum \* (-1))

} else if (сейчас (было перед этим) НЕ минус: tempMinus = false (по умолчанию)) {

if (сейчас (было перед этим) умножение: tempMult=true (включилось первым нажатием на \*)

1. Взять последнее с массива в saveMult удаляя с массива
2. Умножить последнее с массива на сейчас-введенное и занести в массив
3. Сбросить состояние умножения
4. Сделать состояние деления
5. saveMult = numbers.pop()
6. numbers[numbers.length] = (parseInt(saveMult) / parseInt(temp))
7. tempMult = false
8. tempMinus = true

} else if (сейчас (было перед этим) деление: tempSplit=true (включилось первым нажатием на /)

1. Взять последнее с массива в saveSplit удаляя с массива
2. Поделить последнее с массива на сейчас-введенное и занести в массив
3. Сбросить состояние деления
4. Сделать состояние умножения
5. saveSplit = numbers.pop()
6. numbers[numbers.length] = (parseInt(saveSplit) / parseInt(temp))
7. tempSplit = false
8. tempMinus = true

} else {

1. Получить предыдущее значение в массив
2. Поставить минус
3. Включить глобальное состояние «минуса»
4. numbers.[numbers.length] = saveNum
5. title.innerText + ‘-’
6. tempMinus = true

}

**Плюс**: что делать при нажатии

Плюс: функция () {

if (сейчас (было перед этим) умножение: tempMult=true (включилось первым нажатием на \*)

1. Взять последнее с массива в saveMult удаляя с массива
2. Умножить последнее с массива на сейчас-введенное и занести в массив
3. Сбросить состояние умножения
4. Сделать состояние деления
5. saveMult = numbers.pop()
6. numbers[numbers.length] = (parseInt(saveMult) / parseInt(temp))
7. tempMult = false
8. tempMinus = true

} if (сейчас (было перед этим) деление: tempSplit=true (включилось первым нажатием на /)

1. Взять последнее с массива в saveSplit удаляя с массива
2. Поделить последнее с массива на сейчас-введенное и занести в массив
3. Сбросить состояние деления
4. Сделать состояние умножения
5. saveSplit = numbers.pop()
6. numbers[numbers.length] = (parseInt(saveSplit) / parseInt(temp))
7. tempSplit = false
8. tempMult = true

} else if (сейчас (было перед этим) минус: tempMinus=true (включилось первым нажатием на -)

1. Значение в темпе сделать отрицательным и занести в массив
2. Сбросить состояние минуса
3. Сделать глобальное состояние умножения
4. numbers[numbers.length] = (saveNum \* (-1))
5. tempMinus = false
6. tempMult - true

} else {

1. Предыдущее значение занести в массив
2. Добавить минус к выводу
3. numbers[numbers.length] = saveNum
4. title.innerText + ‘+’

}

**РАВНО** Нужно идти от умножения до минуса

Равно: функция () {

if (сейчас умножение: tempMult = true) {

1. 1) Взять последнее с массива в saveMult удаляя с массива
2. Умножить последнее с массива на сейчас-введенное и занести в массив
3. Вывести в output
4. Сбить индикатор умножения (и все остальные)

1) saveMult = numbers.pop()

2) numbers[numbers.length] = (parseInt(tempMultNum) \* parseInt(saveNum))

3) title.innerText = sum

4) tempMult = false

tempSplit = false

tempMinus = false

} else if (сейчас деление: tempSplit=true)

1. Взять последнее с массива в saveSplit удаляя с массива
2. Поделить последнее с массива на сейчас-введенное и занести в массив
3. Цикл сложения
4. Вывести в output
5. Сбить индикатор умножения (и все остальные)
6. saveSplit = numbers.pop()
7. numbers[numbers.length] = (parseInt(saveSplit) / parseInt(temp))
8. for (var i=0; i<numbers.length; i++){

sum = sum + parseInt(numbers[i]);

}

1. title.innerText = sum
2. tempMult = false

tempSplit = false

tempMinus = false

} else if (сейчас минус: tempMinus=true) {

1. значение в темпе сделать отрицательным и занести в массив
2. Цикл сложения
3. Вывести в output
4. Сбить индикатор минуса (и все остальные)
5. numbers[numbers.length] = (saveNum \* (-1))
6. for (var i=0; i<numbers.length; i++){

sum = sum + parseInt(numbers[i]);

}

1. title.innerText = sum
2. tempMult = false

tempSplit = false

tempMinus = false

}

}