జావా స్క్రిప్ట్

4.0 పరిచయం

4.1 జావాస్క్రిప్ట్‌లో మూలకాంశ ప్రాప్యత

4.2 చరాంశాలు, ఆపరేటర్‌లు మరియు వ్యాఖ్యలు

4.3 పంక్షన్‌లు, ఆబ్జెక్ట్‌లు మరియు ఈవెంట్‌లు

4.4 స్ట్రింగ్‌లు, సంఖ్యలు, గణిత మరియు తేదీ పంక్షన్‌లు

4.5 అర్రేలు

4.6 బూలియన్‌లు, నిబంధనలు మరియు లూప్‌లు

4.7 సాధారణ వ్యక్తీకరణలు (RegEx)

4.0 పరిచయం

ఎందుకు

జావాస్క్రిప్ట్ సవివరమైన స్క్రిప్టింగ్ భాష. జావాస్క్రిప్ట్ కోడ్‌లో ప్రతి సేకరణ కేవలం వరుస పంక్తులుగా వ్రాసే శ్రేణి, ఈ పంక్తులు సెమీకోలన్‌తో ముగుస్తాయి. దీన్ని స్క్రిప్ట్ అంటారు. ఒక జావాస్క్రిప్ట్ పత్రంలో పొందుపరిచిన స్క్రిప్ట్‌లు ఎన్నయినా ఉండవచ్చు.

జావాస్క్రిప్ట్ కేస్ ఆధారిత స్క్రిప్టింగ్ భాష - అంటే ఇది సంకలనం చేయబడదు (జావాకు భిన్నమైనది), ఇది అమలు సమయంలో నిర్వహించబడుతుంది.

చాలా పంక్తులను కలిపితే స్క్రిప్ట్ రూపొందుతుంది. కోడ్ యొక్క కొంత భాగం ధనుర్బంధ బ్రాకెట్‌లను {} ఉపయోగించి పరిమితం చేయబడుతుంది.

జావాస్క్రిప్ట్ మరియు జావా

జావాస్క్రిప్ట్ మరియు జావా ఇంచుమించుగా ఒకేలా కనిపించినప్పటికీ, అవి చాలా భిన్నమైనవి. అత్యంత ప్రాథమిక భేదం జావాస్క్రిప్ట్ అంశం ఆధారిత సాఫ్ట్‌వేర్ అభివృద్ధి పదసమాహారానికి మద్దతివ్వదు. జావా అనేది జావాస్క్రిప్ట్ చరాంశాలు ప్రత్యేకంగా టైప్ చేయబడినప్పుడు (దీని గురించి తర్వాత మరింత) అత్యంత ఎక్కువగా టైప్ చేయబడే భాష. అతిముఖ్యంగా, జావాస్క్రిప్ట్ ఆబ్జెక్ట్‌లు గతిశీలమైనవి - అమలు సమయంలో డేటా సభ్యుల సంఖ్య మరియు ఆబ్జెక్ట్ పద్ధతులు మారవచ్చు.

ఎలా

జావాస్క్రిప్ట్ అనుకలన భాగంగా నియోగించేలా <style> మూలకాంశంలో లేదా వేరొక ఫైల్‌లో ఉపయోగించవచ్చు. మీరు మీ కోడ్ అనుకలన భాగంగా నియోగించేలా వ్రాయవచ్చు, కానీ ప్రతి బటన్ కోసం మీరు కోడ్ మొత్తాన్ని పూర్తిగా వ్రాయాల్సి ఉంటుంది, కనుక ఇది అంత ఎక్కువగా సహాయకరం కాదు. స్క్రిప్ట్‌లను వ్రాయడానికి ఉత్తమ మార్గం ముఖ్య శీర్షిక ట్యాగ్‌ల (`<head>...</head>`)లో స్వతంత్ర విధుల ద్వారా వ్రాయడం. మీరు స్క్రిప్ట్‌లను వేరొక ఫైల్‌లో కూడా వ్రాయవచ్చు.

దీన్ని ప్రయత్నించండి
అనుకలన భాగంగా నియోగించే స్క్రిప్ట్‌కు ఇక్కడ ఒక ఉదాహరణ అందించబడింది. ఇది 'బటన్' మూలకాంశాన్ని సృష్టిస్తుంది మరియు క్లిక్ చేసినప్పుడు పాప్ అప్‌ను చూపుతుంది.

```
<input type ="button" id="button1" value=" నన్ను నొక్కండి!"
onclick="alert(' మీరు బటన్‌ను క్లిక్ చేసారు!');" />
```

ఆన్‌క్లిక్ ఈ బటన్‌ను క్లిక్ చేసినప్పుడు జరగవలసిన చర్యను సూచిస్తుంది.

> హెచ్చరిక: ఈవెంట్ లక్షణాలను ఉపయోగించకూడదు. అవి HTMLని జావాస్క్రిప్ట్‌తో మిళితం చేస్తాయి, దీని వలన పత్రాలను చదవడం మరియు డీబగ్ చేయడం కష్టతరం అవుతుంది మరియు ఇది అధిక పరిమాణంలో ఉంటుంది.

మీరు శైలి మూలకాంశంలో అదే జావాస్క్రిప్ట్ కోడ్‌ను ఎలా వ్రాయాలో ఇక్కడ చూపబడింది.

```
<head>
  <script>
              function doSomething() {
                           alert(' మీరు బటన్‌ను క్లిక్ చేసారు!') ;
              }
      var button1 = document.getElementById('button1');
      button1.onclicked = doSomething;
  </script>
</head>

<input type ="button" id="button1" value=" నన్ను నొక్కండి!" />
```

దీని పనితీరు మేము ఎగువ చూసిన దానికి సారూప్యంగా ఉంటుంది, అది మినహా ఇది కోడ్‌ను వ్రాయడంలో మరింత నిర్మాణక్రమ మార్గం. ముఖ్యంగా చెప్పాలంటే, ఇది ప్రాధాన్యపరిచిన పద్ధతి!

4.1 జావాస్క్రిప్ట్‌లో మూలకాంశం ప్రాప్యత

ఎందుకు
XHTML ఫారమ్ మూలాకాంశంతో అనుబంధించిన ఆబ్జెక్ట్‌ను జావాస్క్రిప్ట్‌లో పేర్కొనడానికి అనేక మార్గాలు ఉన్నాయి. డాక్యుమెంట్ ఆబ్జెక్ట్ మోడల్ (DOM) అనేది ఫారమ్‌లను మరియు డాక్యుమెంట్ ఆబ్జెక్ట్ యొక్క మూలకాంశ అర్రేలను ఉపయోగించడానికి అనుసరించే ఒక మార్గం, ఇది విండో ఆబ్జెక్ట్ ద్వారా సూచించబడుతుంది.
పునరుద్ఘాటించేందుకు, విండో ఆబ్జెక్ట్ అంటే XHTML కంటెంట్‌ను ప్రదర్శించే విండో. ప్రతి విండో పత్రం అనే లక్షణాన్ని కలిగి ఉంటుంది. ప్రతి డాక్యుమెంట్ ఆబ్జెక్ట్ ఫారమ్‌ల అర్రేను కలిగి ఉంటుంది మరియు ప్రతి మూలకాంశం పత్రం యొక్క ఫారమ్‌ను సూచిస్తుంది. ఫారమ్‌లు అర్రే మూలకాంశంలో బటన్‌లు, మెనులు మొదలైనటువంటి XHTML ఫారమ్ మూలకాంశాలను సూచించే ఆబ్జెక్ట్‌లను కలిగి ఉండే మూలకాంశం అర్రేని లక్షణం వలె కలిగి ఉంటుంది.

ఎలా

పత్రం యొక్క మూలకాంశాలు డేటా మరియు కార్యకలాపాలు రెండింటిని కలిగి ఉండే ఆబ్జెక్ట్‌లు.డేటాను లక్షణాలుగా , మరియు కార్యకలాపాలను పద్ధతులుగా పిలుస్తారు.

ఈ ఉదాహరణను ఒకసారి చూడండి:

```
<input type="text" name="age">
```

ఇక్కడ, రకం మరియు పేరు ఒక ఆబ్జెక్ట్ యొక్క లక్షణాలుగా మరియు 'వచనం' మరియు 'వయస్సు' దాని విలువలుగా ఉన్నాయి.

DOM చిరునామా అంటే జావాస్క్రిప్ట్ ఆబ్జెక్ట్ యొక్క చిరునామా. మీరు డాట్ లక్షణాన్ని ఉపయోగించి DOM మూలకంలోని లోతైన మూలకాంశాలను ప్రాప్యత చేయగలరు. మూలకాంశం యొక్క పేరు లేదా విశిష్ట id దాన్ని సూచించేందుకు ఉపయోగించబడుతుంది. దీన్ని getElementByName() లేదా getElementById() వంటి పద్ధతులను ఉపయోగించడం ద్వారా చేయవచ్చు.

ఈ ఉదాహరణను ఒకసారి చూడండి:

```
var dom = document.getElementById("button1");
```

ఈ కోడ్ పంక్తి *button1*ను విశిష్ట గుర్తింపు పేరుగా లేదా idగా కలిగి ఉన్న మూలకాంశం ప్రాప్యత చేసేందుకు ఉపయోగించబడుతుంది, ఇది పత్రంలో ఒక భాగంగా ఉంటుంది.

దీన్ని ప్రయత్నించండి

<u>ఒక ఫారమ్‌లో టిక్ చేయబడిన ఎంపిక పెట్టెల సంఖ్యను లెక్కించడానికి:</u>

ఈ స్క్రిప్ట్ ఎంపిక పెట్టె ద్వారా 3 అంశాలు, గల సూచనప్రాయమైన అర్రేను సృష్టిస్తుంది. అన్ని మూలకాంశాలు ఒకే పేరు మరియు idని కలిగి ఉంటాయి, దీని వలన స్క్రిప్ట్‌లో వాటిని తర్వాత సులభంగా ప్రాప్యత చేయవచ్చు. మూలకాంశాలకు వ్యక్తిగత చిరునామాలు అర్రే సూచీలను ఉపయోగించి నిల్వ చేయబడతాయి, ఇవి 0తో మొదలై మరియు -1 పొడవు వరకు వెళతాయి.

ఇన్‌పుట్ రకం చెక్‌బాక్స్ లాంటిది కావడం వలన, ప్రతి మూలకాంశం కూడా 'ఎంచుకోబడింది' అనే అదనపు లక్షణాన్ని కలిగి ఉంటుంది. ఎంచుకొని ఉంటే ఎంచుకున్న లక్షణం ఒప్పుకు సెట్ చేయబడుతుంది, మరియు సెట్ చేయబడనట్లయితే తప్పుకు సెట్ చేయబడుతుంది.

```
<form id="FavoriteColor">
  <input type="checkbox" name="color"
      value="red" />  ఎరుపు
  <input type="checkbox" name="color"
      value="blue" />  నీలం
  <input type="checkbox" name="color"
      value="green" />  ఆకుపచ్చ
</form>


var numChecked = 0'
var dom = document.getElementById("FavoriteColor");
for (i = 0; i < dom.color.length; i++)
      if (dom.car[i].checked)
            numChecked++;
```

4.2 చరాంశాలు, ఆపరేటర్‌లు మరియు వ్యాఖ్యలు

4.2.1 చరాంశాలు

ఎందుకు

విలువలను కలిగి ఉండే చరాంశాలను సులభంగా అందించండి. ఒక చరాంశాన్ని ఉపయోగించే ముందు, మీరు దాన్ని నిర్ధారించాల్సి ఉంటుంది.
జావాస్క్రిప్ట్‌లో ఒక చరాంశం అనేది ఒక ఆబ్జెక్ట్‌కు విలువగా లేదా సూచనగా ఏవిధంగానైనా ఉపయోగించగలిగే గతిశీలమైన టైపింగ్ భాష.
ఒక చరాంశానికి విలువ కేటాయించబడినట్లయితే, అది స్వయంచాలకంగా ఆ విలువ యొక్క రకంగా తీసుకుంటుంది.
జావాస్క్రిప్ట్ రకాలు 2 సమూహాలుగా విభజించబడ్డాయి:
ఆదిమ రకాలు: సంఖ్య, స్ట్రింగ్, బూలియన్, నిర్వచించబడనివి మరియు శూన్యం
సూచిత రకాలు: ఆబ్జెక్ట్‌లు, అర్రేలు మరియు ఫంక్షన్‌లు

చరాంశాల గురించి గుర్తుంచుకోవలసిన ముఖ్యమైన విషయాలు:

- జావాస్క్రిప్ట్ HTML మరియు CSS వలె కాకుండా కేస్ ఆధారితమైనది
- కీలక పదాలు (ఉదాహరణకు var, document మొదలైనవి) చరాంశాల పేర్లుగా ఉపయోగించలేరు
- ఒక చరాంశం యొక్క డిఫాల్ట్ విలువ (పేర్కొనకపోయినట్లయితే) నిర్వచించబడదు
- స్ట్రింగ్‌లు కొటేషన్స్‌లో ప్రకటించాలి

చరాంశం పరిధి:

- స్థానిక చరాంశాలు:
  ఒక చరాంశం ఒక ఫంక్షన్‌లో ప్రకటించబడినప్పుడు, అది ఆ ఫంక్షన్‌లో మాత్రమే ప్రాప్యత కలిగి ఉంటుంది మరియు ఫంక్షన్ ముగిసాక అది చెల్లుబాటు కాదు.
- సార్వత్రిక చరాంశాలు
  ఒక చరాంశం స్క్రిప్ట్ ఫైల్ యొక్క పైస్థాయిలో (ఫంక్షన్‌కు వెలుపల) ప్రకటించబడినప్పుడు, అది జావాస్క్రిప్ట్ ప్రోగ్రామ్ మొత్తంలో ఎక్కడైనా ప్రాప్యత చేయబడుతుంది.
- ప్రకటించబడని చరాంశాలు స్వయంచాలకంగా సార్వత్రిక చరాంశాలుగా చెప్పవచ్చు

ఎలా

ఒక చరాంశానికి విలువను దాని విలువను (ఇంటర్‌ప్రిటర్ సూచనప్రాయంగా దాన్ని చరాంశంగా ప్రకటించే సందర్భంలో) కేటాయించడం ద్వారా లేదా `var` కీలక పదంతో మొదలుపెట్టే నిర్ధారణ ప్రకటనలో జాబితా చేయడం ద్వారా ఒక విలువను కేటాయించవచ్చు.

దీన్ని ప్రయత్నించండి

```
var counter,
    pi = 3.14,
    flag = true,            // బూలియన్ చరాంశం
    shape = “circle";       //స్ట్రింగ్ చరాంశం
```

4.2.2 ఆపరేటర్‌లు

ఎందుకు

విలువలపై చేయదగిన కూడికలు, తీసివేతలు మొదలైనవి అమలు చేయాల్సిన వివిధ రకాలైన ఫంక్షన్‌లు ఉన్నాయి. ఈ ఫంక్షన్‌లు ఆపరేటర్‌లను ఉపయోగించి అమలు చేయబడతాయి. కొన్ని ఆపరేటర్‌లు + వంటివి స్ట్రింగ్‌ల జోడింపు కోసం కూడా ఉపయోగించవచ్చు.

ఎలా

జావాస్క్రిప్ట్‌లో వివిధ రకాలైన ఆపరేటర్‌లు ఉన్నాయి.

- అంకగణితం
  - ఇది ప్రాథమిక అంకగణిత క్రియలను నిర్వర్తిస్తుంది.

+	కూడిక
-	తీసివేత
*	గుణకారం
/	భాగహారం
%	శేషం ఫంక్షన్
+ +, --	ఒక చరాంశం పెరుగుదల/తరుగుదల (ఏకాంశ ఆపరేటర్)

+ ఒక జోడింపు ఆపరేటర్‌గా కూడా ఉపయోగించవచ్చు.

```
var a = "Hello" + "World"
```

దీని అర్థం *a* *Hello World* కలిగి ఉంటుంది.

- సమానత్వం
  - ఇది 2 విలువలను మరియు రకాలను సరిపోల్చడానికి ఉపయోగపడుతుంది

=	కేటాయింపు ఆపరేటర్ `var a = 2` a అనే చరాంశానికి 2 అనే విలువను కేటాయిస్తుంది
= =	సమానత్వం ఆపరేటర్ `var a = "2" == 2;` //ఒప్పు ఇది రెండు చరాంశాల యొక్క రకాలను మరియు విలువలను ఎంపిక చేస్తుంది

- సరిపోల్చడం
  - ఇది 2 విలువలను సరిపోల్చేందుకు ఉపయోగించబడుతుంది. ఇది స్ట్రింగ్‌లను సరిపోల్చేందుకు కూడా ఉపయోగించవచ్చు.

< , >	కంటే తక్కువ, కంటే ఎక్కువ
< = , > =	కంటే తక్కువ లేదా దానికి సమానం, కంటే ఎక్కువ లేదా దానికి సమానం

4.2.3 వ్యాఖ్యలు

ఎందుకు

వ్యాఖ్యలు అనగా కోడ్‌లో కంపైలర్ ద్వారా నిర్వహించబడని వ్యక్తులు చదవగలిగే పంక్తులు.
అవి ఎటువంటి నిర్దిష్ట ఆకృతిని అనుసరించనవసరం లేదు. ఇవి కోడ్ భాగం గురించి అర్థం చేసుకోవడంలో ప్రోగ్రామర్‌కు సహాయకరంగా ఉండటానికి మాత్రమే ఉపయోగించబడతాయి.

ఎలా

ఏక పంక్తి వ్యాఖ్యలు //తో వ్రాయవలసి ఉంటుంది
బహుళ పంక్తుల వ్యాఖ్యలు /* మరియు */ లోపల మాత్రమే ఉండవలసి ఉంటుంది

దీన్ని ప్రయత్నించండి

var a = 2 ; //ఇది ఒక వ్యాఖ్య. ఇది ఒక పంక్తిలో మాత్రమే ఉండాలి.
var b = “Hello" ; /*ఇది బహుళ-పంక్తుల వ్యాఖ్య.
ఇది ఒక పంక్తి కంటే ఎక్కువగా వ్యాప్తి చెందుతుంది.*/

4.3 ఫంక్షన్‌లు, ఆబ్జెక్ట్‌లు మరియు ఈవెంట్‌లు

4.3.1 ఫంక్షన్‌లు

ఎందుకు

ఫంక్షన్ అనేది ఒక నిర్దిష్ట కార్యాచరణను నిర్వర్తించే కోడ్ యొక్క నిశ్చల మాడ్యూల్.
ప్రత్యేకించి కూడికలు, వినియోగదారును ఇన్‌పుట్ కోసం అడగడం వంటి ఒకే రకమైన పనులను పదేపదే చేస్తున్నట్లయితే, ఇది మీ కోడ్ యొక్క పునర్వినియోగాన్ని మెరుగుపరుస్తుంది.

ఎలా

ఒక ఫంక్షన్ నిర్వచనం ఆ ఫంక్షన్ యొక్క మరియు గా పిలువబడే ఒక సమ్మేళన ప్రకటనల సమితిని కలిగి ఉంటుంది.
ఫంక్షన్ శీర్షిక రిజర్వ్ చేయబడిన (కీలకపదం) ఫంక్షన్‌తో ప్రారంభమవుతుంది. ఇది ఫంక్షన్ యొక్క ప్రయోజనాన్ని వివరించే ఒక విశిష్ట పేరును కూడా కలిగి ఉంటుంది. ప్రధాన విషయం ఫంక్షన్ ద్వారా నిశ్చయించబడిన స్టేట్‌మెంట్‌లను కలిగి ఉంటుంది.
జావాస్క్రిప్ట్ ఫంక్షన్‌లు ఆబ్జెక్ట్‌లుగా ఉంటాయి, అందువలన వాటిని సూచించే చలరాశులు ఇతర ఆబ్జెక్ట్ సూచికలుగా భావించబడతాయి.

ఫంక్షన్‌లు పరామితులను కూడా కలిగి ఉంటాయి. పరామితులు ఫంక్షన్ శీర్షికలో విలువలను కలిగి ఉండగల చలరాశులు.ఒక ఫంక్షన్ పిలువబడినప్పుడు, ఫంక్షన్‌కు జారీ చేయబడతాయి, దీని అర్థం విలువలు ఫంక్షన్‌కు ఇన్‌పుట్‌గా జారీ చేయబడతాయి. దీన్ని అప్పుడు ఫంక్షన్ లోపల వినియోగించవచ్చు. గుర్తుంచుకోవలసిన ఒక ముఖ్యమైన విషయం ఏమిటంటే ఫంక్షన్ పరామితులు కాల్-ద్వారా-విలువ పద్ధతిని ఉపయోగించి జారీ చేయబడుతుంది. దీని అర్థం విలువ మాత్రమే (ఆబ్జెక్ట్ యొక్క సూచన కాదు) ఫంక్షన్‌కు ఇన్‌పుట్‌గా జారీ చేయబడుతుంది.

```
function functionName(var1, var2) {
              /* function specific code here
              …
              */
              Return returnVal; // optional return value
}
```

దీన్ని ప్రయత్నించండి

వినియోగదారు నుండి 2 స్ట్రింగ్‌లను అంగీకరించిన తర్వాత వాటిని ఒక జతపరచబడిన స్ట్రింగ్‌గా అందించేలా ఒక ఫంక్షన్‌ను వ్రాయండి.

```
function returnConcatString(var1 var2) {
              var result = "";                              // ఒక ఖాళీ
స్ట్రింగ్‌ను అవుట్‌పుట్‌గా అందించేలా చేస్తోంది
              for (var i = 0; i < arguments.length; i++) {
                            result = result + " " + arguments[i];
              }
      return result;
}

var b = returnConcatString("Hello", "World");
```

b *Hello World* కలిగి ఉంటుంది.

4.3.2 ఆబ్జెక్ట్‌లు

ఎందుకు

ఒక ఆబ్జెక్ట్ అనేది లక్షణాల యొక్క సమాహారం. లక్షణం యొక్క విలువ ఫంక్షన్ అయినట్లయితే, అప్పుడు అది పద్ధతిగా పిలువబడుతుంది.

ఒక ఆబ్జెక్ట్ తప్పనిసరిగా 2 విషయాలను కలిగి ఉండాలి: ఒక లక్షణం మరియు ఒక పద్ధతి.

దాన్ని సులభతరం చేసేందుకు, ఒకే రకానికి చెందిన అన్ని ఆబ్జెక్ట్‌లు ఒకే రకమైన లక్షణాలను కలిగి ఉంటాయి, కానీ వాటి లక్షణం విలువలు మారవచ్చు.

ఎలా

ఒక ఆబ్జెక్ట్ దానితో అనుబంధమై ఉన్న లక్షణాలు మరియు పద్ధతులను నిర్వచించడం ద్వారా సృష్టించబడుతుంది. మీరులక్షణాలు మరియు పద్ధతులను డాట్ ఆపరేటర్ (.) ఉపయోగించడం ద్వారా ప్రాప్యత చేయగలరు

ఒక ఆబ్జెక్ట్ యొక్క లక్షణాలను ఈ విధంగా ప్రాప్యత చేయవచ్చు:

```
objectName.propertyName ;
        objectName["propertyName"] ;
```

ఒక ఆబ్జెక్ట్ యొక్క పద్ధతులను ఈ విధంగా ప్రాప్యత చేయవచ్చు:

```
objectName.methodName() ;
```

ఒక ఖాళీ ఆబ్జెక్ట్ ఇటువంటి ఒక కన్స్ట్రక్టర్‌ను ఉపయోగించి ప్రకటించబడుతుంది:

```
var car = new Object () ;
```

మీరు ఆబ్జెక్ట్‌ను సృష్టించిన తర్వాత, లక్షణాలు ప్రారంభించబడతాయి.

```
car.type = "Sedan";
car.color = "red";
```

దీన్ని ప్రయత్నించండి

కారు వంటి ఆబ్జెక్ట్‌ను రకం, రంగు మరియు మోడల్ వంటి వివిధ లక్షణాలతో సృష్టించండి.

```
var car = {type: 'Sedan',
                    color: 'red',
                    Model: 'Fiat};
```

4.3.3 ఈవెంట్‌లు

ఎందుకు

ఈవెంట్ అనేది వినియోగదారు చర్య లేదా ఏదో జరిగిందని తెలియజేసే ఒక నోటిఫికేషన్. జావాస్క్రిప్ట్ కోడ్ ఈవెంట్‌ల ద్వారా HTML పేజీలతో ప్రతిస్పందిస్తుంది. ఉదాహరణకు, ఒక బటన్‌ను క్లిక్ చేయడం, మీ మౌస్‌ను ఏదైనా వచనంపై కదిలించడం మొదలైనవి ఈవెంట్‌లకు ఉదాహరణగా చెప్పవచ్చు.

ఎలా

ఒక ఈవెంట్ సంభవించినప్పుడు దాన్ని సూచనప్రాయంగా పిలిచే స్క్రిప్ట్‌ను ఈవెంట్ హ్యాండ్లర్ అంటారు.ప్రతి మూలకాంశం దాని అనుబంధిత ఈవెంట్ లక్షణాన్ని కలిగి ఉంటుంది. చాలా మూలకాంశాలు ఒకే రకమైన ఈవెంట్ మూలకాంశాలను కలిగి ఉంటాయి.

దీని అర్థం `onclick` వంటి ఒక ఈవెంట్ ఒక లింక్ (యాంకర్ ట్యాగ్) మరియు ఒక వచన పెట్టెలో అవే చర్యలను కలిగి ఉంటుంది.

కొన్ని ఈవెంట్‌లు మరియు వాటి ట్యాగ్ లక్షణాలు ఇక్కడ ఉన్నాయి:

ఈవెంట్	ట్యాగ్ లక్షణం
blur	onblur
click	onclick
dblclick	ondblclick

focus	onfocus
keydown	onkeydown
keypress	onkeypress
keyup	onkeypress
load	onload
mousedown	onmousedown
mousemove	onmousemove
mouseover	onmouseover
reset	onreset
select	onselect
submit	onsubmit
unload	onunload

దీన్ని ప్రయత్నించండి

పత్రం యొక్క ప్రధాన విషయం లోడ్ చేయబడినప్పుడు, ఒక హెచ్చరిక సందేశాన్ని అందిస్తుంది.

```
<html>
<head>
              <title> load.html </title>
              <script type = "text/javascript"  src = "load.js">
              </script>
</head>
<body onload = load_greeting();">
              <p> పేజీ లోడ్ అయింది! </p>
</body>
</html>

//load.js
function load_greeting() {
              alert(" ఇది శుభాకాంక్షలు తెలిపే ఒక పాప్ అప్ సందేశం - హాయ్! నా పేజీకి
స్వాగతం!");
}
```

4.4 స్ట్రింగ్‌లు, సంఖ్యలు, గణిత మరియు తేదీ ఫంక్షన్‌లు

4.4.1 స్ట్రింగ్‌లు

ఎందుకు

జావాస్క్రిప్ట్ స్ట్రింగ్‌లు అక్షరాల అర్రేల వలె నిల్వ చేయబడలేదు లేదా భావించబడలేదు, బదులుగా అవి యూనిట్ రాశి ప్రమాణాలుగా ఉంటాయి.

ఎలా

స్ట్రింగ్‌లు నిర్వహించవలసిన చాలా ఫంక్షన్‌లు లేదా కార్యకలాపాలను కలిగి ఉండవచ్చు.
విలువలు ఆబ్జెక్ట్‌ల వలే, స్ట్రింగ్ పద్ధతులు స్ట్రింగ్ రాశి విలువలుగా ఉపయోగించవచ్చు.
ఉదాహరణకు, స్ట్రింగ్‌లోని అక్షరాల సంఖ్య పొడవు లక్షణంలో నిల్వ చేయబడుతుంది.

సాధారణ స్ట్రింగ్ పద్ధతులలో కొన్ని క్రింది పట్టికలో చూపబడతాయి.

పద్ధతి	పరామితి	ఫలితం
length	ఏదీ కాదు	స్ట్రింగ్ ఆబ్జెక్ట్‌లోని అక్షరాల సంఖ్యను అందిస్తుంది
charAt	ఒక సంఖ్య	స్ట్రింగ్ ఆబ్జెక్ట్‌లో పేర్కొన్న స్థానంలో ఉన్న అక్షరాన్ని అందిస్తుంది
indexOf	ఒక-అక్షరం స్ట్రింగ్	స్ట్రింగ్ ఆబ్జెక్ట్‌లో పరామితి యొక్క స్థానాన్ని అందిస్తుంది
substring	రెండు సంఖ్యలు	స్ట్రింగ్ ఆబ్జెక్ట్ యొక్క ఉప స్ట్రింగ్‌ను మొదటి పరామితి స్థానం నుండి రెండవ పరామితికి అందిస్తుంది
toLowerCase	ఏదీ కాదు	ఏవైనా పెద్ద అక్షరాలను చిన్న అక్షరాలుగా మార్చుతుంది
toUpperCase	ఏదీ కాదు	ఏవైనా చిన్న అక్షరాలను పెద్ద అక్షరాలుగా మార్చుతుంది

ఈ పద్ధతులకు అదనంగా, ఎస్కేప్ అక్షరాలుగా పిలువబడే కొన్ని నిర్దిష్ట అక్షర క్రమాలు కూడా ఉపయోగించబడుతాయి.

\n	కొత్త పంక్తి
\t	ట్యాబ్ (8 అక్షరాల ఖాళీ)
\r	ఎంటర్

దీన్ని ప్రయత్నించండి

```
var str = “Hello World" ;
```

```
str.charAt(2);                          //  అవుట్‌పుట్ l
str.indexOf('o');                //  అవుట్‌పుట్ 4
str.substring(2,4);                //  అవుట్‌పుట్ llo
str.toLowerCase();                //  అవుట్‌పుట్ hello world
```

4.4.2 సంఖ్యలు

ఎందుకు

నంబర్ ఆబ్జెక్ట్ ఉపయోగకరమైన లక్షణాలతో కూడిన ఒక సేకరణను కలిగి ఉంటుంది.

ఎలా

క్రింది పట్టిక సంఖ్యా లక్షణాలను సూచిస్తుంది.

లక్షణం	అర్థం
MAX_VALUE	అతి పెద్ద సూచించదగిన సంఖ్య
MIN_VALUE	అతి చిన్న సూచించదగిన సంఖ్య
NaN	సంఖ్య కాదు
POSITIVE_INFINITY	అనంతాన్ని సూచించే ప్రత్యేక విలువ
NEGATIVE_INFINITY	ప్రతికూల అనంతాన్ని సూచించే ప్రత్యేక విలువ
PI	pi యొక్క విలువ (3.14)

4.4.3 గణిత ఫంక్షన్

ఎందుకు

గణిత ఆబ్జెక్ట్ సైన్ మరియు కొసైన్ వంటి త్రికోణమితి ఫంక్షన్‌ల కోసం పద్ధతులను కలిగి ఉంది. ఇది సాధారణంగా వినియోగించే ఇతర గణిత ఫంక్షన్‌ల కోసం కూడా పద్ధతులను కలిగి ఉంటుంది.

ఎలా

గణిత ఫంక్షన్ సైన్ (సైన్) మరియు కాస్ (కొసైన్) వంటి త్రికోణమితి ఫంక్షన్‌లను, దానితో పాటుగా సంఖ్యను కుదించే `floor`; ఒక సంఖ్యను సమీపంలోని పూర్ణ విలువకు పూరించే `round`; రెండు నంబర్‌లలోని పెద్దదాన్ని అందించే `max` మొదలైన నిర్దిష్ట గణిత ఫంక్షన్‌లను కూడా కలిగి ఉంటుంది.
అన్ని గణిత ఫంక్షన్‌లు Math ఆబ్జెక్ట్‌ను ఉపయోగించి సూచించబడతాయి.

దీన్ని ప్రయత్నించండి

```
Math.floor(4.2343);                //అవుట్‌పుట్ 4
```

4.4.4 తేదీ ఫంక్షన్

ఎందుకు
తేదీ ఫంక్షన్ తేదీ మరియు సమయాన్ని సూచించేందుకు ఉపయోగించబడుతుంది.

ఎలా
తేదీ ఆబ్జెక్ట్‌ను ఉపయోగించడం ద్వారా తేదీ ఆబ్జెక్ట్ సూచించబడుతుంది.
దీన్ని ఈ స్టేట్‌మెంట్‌ను ఉపయోగించి సృష్టించవచ్చు:

```
var today = new Date() ;
```

ఇక్కడ తేదీ ఆబ్జెక్ట్‌ను ఉపయోగించే కొన్ని సాధారణ పద్ధతులు ఉన్నాయి.

పద్ధతి	ఫలితం
toLocaleString	తేదీ సమాచారం యొక్క స్ట్రింగ్
getDate	నెలలోని రోజు
getMonth	సంవత్సరంలోని నెల (0 నుండి 11)
getDay	వారంలోని రోజు (0 నుండి 6)
getFullYear	సంవత్సరం
getTime	01-01-1970 నుండి మిల్లీ సెకన్ల సంఖ్య
getHours	గంట సంఖ్య (0 నుండి 23)
getMinutes	నిమిషం యొక్క సంఖ్య (0 నుండి 59)
getSeconds	సెకను యొక్క సంఖ్య (0 నుండి 59)

4.5 అర్రేలు

ఎందుకు
అర్రే అనేది అదే పేరుతో సూచించబడే విలువల యొక్క సమాహారం. ప్రతి అర్రే మూలకాంశం దాని యొక్క సూచిక ద్వారా ప్రాప్యత చేయబడుతుంది. సూచికలు 0 నుండి మొదలై పొడవు-1 వరకు ఉంటాయి.
జావాస్క్రిప్ట్‌లో గుర్తుంచుకోవలసిన ఒక ముఖ్యమైన విషయం, ఇతర ప్రోగ్రామింగ్ భాషల వలె కాకుండా, అర్రేలు మిశ్రమ రకాన్ని కలిగి ఉంటాయి. దీని అర్థం అర్రేలోని అన్ని మూలకాంశాలు ఒకే రకంలో ఉండనవసరం లేదు.

ఎలా
అర్రే ఆబ్జెక్ట్‌లు, చాలా ఇతర జావాస్క్రిప్ట్ ఆబ్జెక్ట్‌ల వలే కాకుండా, 2 మార్గాల్లో సృష్టించవచ్చు.
మొదటి విధానం, ఆపరేటర్‌ను ఉపయోగించి సాధారణ పద్ధతిలో ఒక ఆబ్జెక్ట్‌ను సృష్టించడం.

```
var list = new Array (1, 2, "three", "fo ur") ;
```

రెండవ విధానం, అప్పటికే ఉన్న ఒక అర్రేను ఉపయోగించడం.

```
var list_2 = [1, 2, "three", "four"];
```

అర్రే విలువ యొక్క కనిష్ట విలువ సున్నాగా ఉంటుంది.

అర్రే మూలకాంశాలు ఒక అర్రే పేరుకు ఉపసర్గగా సంఖ్యాత్మక విలువను ఉపయోగించి ప్రాప్యత చేయబడతాయి. ఒక అర్రే యొక్క పొడవు ఆ అర్రేలోని మూలకాంశాల సంఖ్య. అందువలన, అర్రే లోని చివరి సూచిక ఎల్లప్పుడూ పొడవు-1గా ఉంటుంది.

అర్రేలు వాటితో అనుబంధించబడిన చాలా విధానాలను కలిగి ఉంటాయి. కొన్ని సాధారణమైనవి ఇక్కడ జాబితా చేయబడ్డాయి.

పద్ధతి	ప్రయోజనం
join	ఒక అర్రే ఆబ్జెక్ట్‌లోని అన్ని మూలకాంశాలను స్ట్రింగ్‌కి మార్చుతుంది మరియు వాటిని జోడిస్తుంది
reverse	ఒక అర్రే ఆబ్జెక్ట్‌లోని మూలకాంశాల క్రమాన్ని తల్లక్రిందులు చేస్తుంది
sort	అర్రే ఆబ్జెక్ట్‌లోని అన్ని మూలకాంశాలను ఒక స్ట్రింగ్‌గా మార్చుతుంది మరియు వాటిని అక్షరక్రమం వారీగా క్రమబద్ధీకరిస్తుంది
concat	అర్రే ఆబ్జెక్ట్ చివరన జోడింపు విధానం యొక్క అసలైన పరామితులను జోడిస్తుంది
slice	అర్రే ఆబ్జెక్ట్‌లో పరామితులుగా పేర్కొనబడిన అర్రే యొక్క భాగాన్ని అందిస్తుంది
pop	ఒక అర్రే ఆబ్జెక్ట్ యొక్క ముగింపు భాగం నుండి ఒక మూలకాంశాన్ని తీసివేస్తుంది
push	ఒక అర్రే ఆబ్జెక్ట్ యొక్క ముగింపు భాగానికి ఒక మూలకాంశాన్ని జోడిస్తుంది

దీన్ని ప్రయత్నించండి

ఒక అర్రే ఆబ్జెక్ట్‌ను సృష్టించండి మరియు విలువలను ప్రదర్శించండి. అర్రేకు మరిన్ని మూలకాంశాలను జోడించండి మరియు కొత్త విలువలను ప్రదర్శించండి.

```
//array_example.js
var list = [2, 4, 6];

for (var i = 0; i < list.length; i++) {
              document.write(list[i] + "  ");
}
document.write("<br />");

var new_list = list.concat(8, 10);

for (var j = 0; j < new_list.length; j++) {
              document.write(new_list[j] + "  ");
}
```

అవుట్‌పుట్ ఇలా ఉంటుంది:

2 4 6

2 4 6 8 10

4.6 బూలియన్‌లు, నిబంధనలు మరియు లూప్‌లు

4.6.1 బూలియన్‌లు

ఎందుకు

బూలియన్ అనేది జావాస్క్రిప్ట్‌లో ఒక డేటా రకం. ఇది కేవలం 2 విలువలను మాత్రమే కలిగి ఉంటుంది:
ఒప్పు మరియు తప్పు.

ఎలా

బూలియన్ ఆబ్జెక్ట్‌లు జావాస్క్రిప్ట్‌లోని ఇతర ఆబ్జెక్ట్‌ల వలె ప్రకటించబడతాయి.

```
var bool_t = new Boolean() ;
        bool_t = true;
```

4.6.2 నిబంధనలు

ఎందుకు

షరతులతో కూడిన స్టేట్‌మెంట్‌లు, స్టేట్‌మెంట్‌ల అమలు విధానాన్ని నిర్ధారిస్తాయి కనుక ముఖ్యమైనవి. అవి ఎల్లప్పుడూ బూలియన్ విలువలను అందించే పరామితులను తీసుకుంటాయి. బూలియన్ విలువ ఒప్పు అయితే, అప్పుడు నిబంధనకు వెంటనే ఉన్న స్టేట్‌మెంట్‌ల సమూహాన్ని అమలు పరుస్తుంది.
అవి గణిత సంబంధిత ఆపరేటర్‌లపై ఆధారపడి ఉంటాయి.

ఎలా

స్టేట్‌మెంట్ అమలు విధానం ఆదిమ విలువలు, సంబంధిత వ్యక్తీకరణలు మరియు మిశ్రమ వ్యక్తీకరణలతో కూడిన వాటిపై ఆధారపడగలిగే వ్యక్తీకరణలు. నియంత్రణ వ్యక్తీకరణ ఫలితం ఒప్పు లేదా తప్పుగా ఉంటుంది.
జావాస్క్రిప్ట్ AND, OR మరియు NOT బూలియన్ ఆపరేషన్‌ల వంటి వాటికి కూడా ఆపరేటర్‌లను కలిగి ఉంటుంది.ఇవి && (AND), || (OR) మరియు ! (NOT).

సాధారణంగా రెండు రకాలైన షరతులతో కూడిన స్టేట్‌మెంట్‌లు ఉంటాయి:

1. ఎంపిక స్టేట్‌మెంట్‌లు

   ఇది if-then మరియు if-then-else నిర్మాణంపై ఆధారపడి ఉంటుంది.
   సింటాక్స్ ఈ విధంగా ఉంటుంది:

```
if (condition) {
                /*  షరతు  ఒప్పు  అయినట్లయితే,
    ఈ  స్టేట్‌మెంట్‌లను  అమలు  పరచు
                …
                */   }
                 కానట్లయితే  {
                /*  షరతు  తప్పు  అయినట్లయితే,
    ఈ  స్టేట్‌మెంట్‌లను  అమలు  పరచు
                …
                */   }
```

2. స్విచ్ స్టేట్‌మెంట్

   తనిఖీ చేసేందుకు చాలా షరతులు ఉండి వాటిలో ఒకటి మాత్రమే సరిపోలే పరిస్థితులలో ఇది

ఉపయోగించబడుతుంది. సింటాక్స్ ఈ విధంగా ఉంటుంది:

```
switch (expression) {
                                case value_1:
  // స్టేట్‌మెంట్(లు)
                                case value_2:
  // స్టేట్‌మెంట్(లు)
     ...
     case value_n:                    // స్టేట్‌మెంట్(లు)
     [default:                    // స్టేట్‌మెంట్(లు)]
}
```

స్విచ్ నిర్మాణంలో ఉండే వ్యక్తీకరణ మూల్యాంకనం చేయబడుతుంది మరియు విలువ నిర్మాణంలోని సందర్భాల విలువలతో సరిపోల్చబడతాయి. ఒకటి సరిపోలినట్లయితే, కేస్ విలువపై ఆధారంగా నియంత్రణ స్టేట్‌మెంట్‌లకు బదిలీ చేయబడుతుంది. అమలు ప్రక్రియ ఆపై నిర్మాణం యొక్క అవశేషం కోసం కొనసాగుతుంది.
చాలా సందర్భాలలో, ఒకే ఒక కేస్ స్టేట్‌మెంట్ మాత్రమే అమలు చేయాల్సి ఉంటుంది, వాటి తర్వాత నియంత్రణ స్విచ్ నిర్మాణానికి వెలుపల ఉన్న స్టేట్‌మెంట్‌లకు బదిలీ చేయాలి.
ఆ సందర్భంలో, స్టేట్‌మెంట్ ఉపయోగించవచ్చు.
స్టేట్‌మెంట్ అది సాధారణంగా కనబడే మిశ్రమ స్టేట్‌మెంట్‌కు వెలుపలకు నియంత్రణను బదిలీ చేస్తుంది.

దీన్ని ప్రయత్నించండి
రెండు నంబర్‌లను సరిపోల్చి, వాటిలో పెద్ద నంబర్‌ను అవుట్‌పుట్‌గా అందించేలా ఒక సాధారణ if-else నిర్మాణాన్ని వ్రాయండి.

```
//if_then.js

var a = 5;
var b = 8;

if (a > b)
   document.write("a is greater than b <br/> ");
else
   document.write("a is lesser than b <br/> ");
```

4.6.3 లూప్‌లు

ఎందుకు
ఒక అర్రే లేదా జాబితాలో చాలా రకాలైన మూలకాంశాలను ఒకే రకమైన విధితో నిర్వర్తించేందుకు ప్రాథమికంగా లూప్‌లు ఉపయోగించబడుతాయి. ఆవశ్యకంగా, ఇది నిర్దిష్ట కోడ్ భాగంలో చాలా సార్ల వరకు లూప్ అవుతుంది.

ఎలా
లూప్‌లు కుండలీకరణములలో ఉండే నిర్దిష్ట కోడ్ భాగాన్ని అనుసరించే ఒక సాధారణమైన నియంత్రణ వ్యక్తీకరణను కలిగి ఉంటాయి, సాధారణంగా రెండు రకాలైన లూప్‌లు ఉంటాయి:

1. While లూప్‌లు

నియంత్రణ వ్యక్తీకరణ ఒప్పు కానంత వరకు, While లూప్‌లు నిర్దిష్ట కోడ్ పంక్తుల సమితిని అమలు చేస్తాయిసింటాక్స్ ఈ విధంగా ఉంటుంది:

```
while (conditional statement)
        {
                        /* set of statements
                        …
                        */
        }
```

2. For లూప్‌లు

For లూప్‌లు ఒక ప్రారంభ విలువను, షరతుల విలువను మరియు పెరుగుదల దశను కలిగి ఉంటాయి. చలరాశి యొక్క ప్రారంభ విలువ స్టేట్‌మెంట్‌లు అమలు చేయబడే ప్రతిసారి పెరుగుతూ కొనసాగడం జరుగుతుంది. నియమిత విలువ ఒప్పు కానప్పుడు, లూప్ ముగుస్తుంది. సింటాక్స్ ఈ విధంగా ఉంటుంది:

```
for (initial expression; control express ion; increment
expression)
{
                        /* set of statements
                        …
                        */
        }
```

లూప్‌లు రెండూ విరామం స్టేట్‌మెంట్‌ను ఉపయోగించి డిమాండ్ మేరకు రద్దు ముగియవచ్చు.

దీన్ని ప్రయత్నించండి

1 నుండి 50 వరకు సంఖ్యల లబ్ధాన్ని కనుగొనేందుకు జావాస్క్రిప్ట్ కోడ్‌ను వ్రాయండి.

```
//loops_while.js

var i = 1;
var sum = 0;

while ( i <= 50) {
sum = sum * i ;
}

document.write("The product is" + sum);

//loops_for.js
var sum = 0;
for (var i = 0; i < 51; i++)
sum = sum * i;

document.write("The product is" + sum);
```

4.7 సాధారణ వ్యక్తీకరణలు (RegEx)

ఎందుకు

జావాస్క్రిప్ట్ శక్తివంతమైన నమూనా సరిపోలిక సామర్థ్యాలను సాధారణ వ్యక్తీకరణల ఆధారంగా కలిగి ఉంటుంది. స్ట్రింగ్ ఆబ్జెక్ట్‌ల యొక్క శోధన ఫంక్షన్ ఉపయోగించగలిగినప్పటికీ, RegEx మరింత సమగ్ర కార్యనిర్వాహకతను అందిస్తుంది.

ఎలా

'సాధారణ' అక్షరాలు అంటే మెటా అక్షరాలు కానివి ఉదా., పరిస్థితులపై ఆధారపడి ప్రత్యేక అర్థాలను కలిగి ఉండనివి.
ప్రత్యేక అక్షరాలు లేదా మెటా అక్షరాలకు ఉదాహరణలు:

```
\ | ( ) [ ] { } ^ $ * + ? .
```

మెటా అక్షరాలు వెంటనే ఒక వెనుక స్లాష్(\)తో ప్రారంభమై సాధారణ వ్యక్తీకరణలతో సరిపోలతాయి.

ఒక పూర్ణ విరామం (.) కొత్త పంక్తి మినహా మిగిలిన అన్నింటితోనూ సరిపోలుతుంది. పైన తెలిపిన విధంగా, ఒక స్ట్రింగ్‌లో పూర్ణ విరామాన్ని సరిపోల్చేందుకు, అది ఒక వెనుక స్లాష్‌తో ప్రారంభమవ్వాలి.
ఉదాహరణ:
`/snow./` అనేది “snowy", “snowed", “snows" మొదలైనవాటికి సరిపోలుతుంది

ఒక సాధారణ వ్యక్తీకరణలో మొదటి అక్షరం ఉచ్చారణ సూచక వంకర అడ్డగీత (^) అయితే, అది పేర్కొనబడిన సమితిని త్రోసిపుచ్చుతుంది లేదా విలోమీకరిస్తుంది.
ఉదాహరణ:
[^abc] అనేది 'a', 'b' మరియు 'c' మినహా అన్ని అక్షరాలతో సరిపోలుతుంది

హైఫన్ (-) ఒక పరిధిని పేర్కొంటుంది.

క్రింది పట్టిక పూర్వ నిర్వచిత తరగతుల గురించి మరిన్ని వివరాలను అందిస్తుంది

పేరు	సమానస్థాయి నమూనా	సరిపోలికలు
`\d`	`[0-9]`	ఒక అంకె
`\D`	`[^0-9]`	ఒక అంకె కాదు
`\w`	`[A-Za-z_0-9]`	ఒక అక్షర సంఖ్య
`\W`	`[^A-Za-z_0-9]`	ఒక అక్షర సంఖ్య కాదు
`\s`	`[ \r\t\n\f]`	ఒక తెల్లని ఖాళీ అక్షరం
`\S`	`[^ \r\t\n\f]`	ఒక తెల్లని ఖాళీ అక్షరం కాదు

దీన్ని ప్రయత్నించండి

[\d\.\d\d] //ఒక అంకె, దాని ప్రక్కన పూర్ణ విరామం ఆపై 2 అంకెలతో కూడి ఉండే దానితో సరిపోలుతుంది
[\D\d\D] //ఒక అంకెతో సరిపోలుతుంది

[\d+\.\d*] //ఒకటి లేదా అంతకంటే ఎక్కువ అంకెలు, తర్వాత పూర్ణ విరామం ఆపై 0 లేదా మరిన్ని అంకెలతో సరిపోలుతుంది

[\w\w\w] //3 ప్రక్కప్రక్క పద అక్షరాలతో సరిపోలుతుంది