JavaScript အားအသုံးပြုခြင်း

What is JavaScript?

JavaScript ဟာ Scripting Language တစ်ခုပါ။ Scripting Language တွေဆိုတာ Web တွေခေတ်မစားခင် ဟိုးရှေးရှေးတုန်းတည်းက ရှိခဲ့ပါတယ်။ JavaScript ဟာ scripting language တစ်ခုဖြစ်တာကြောင့် ရိုးရှင်းတဲ့ ပရိုဂရမ်လေးတွေကိုလည်း ရေးသားနိုင်ပါတယ်။

အောက်က JavaScript Is Object-Oriented ဆိုတဲ့ ခေါင်းစဉ်ဟာ ပရိုဂရမ်းမင်းကို အခုမှ စလေ့လာမယ့်သူတွေအတွက် စစခြင်း ဆောက်နဲ့ထွင်းသလိုဖြစ်နေနိုင်ပါတယ်။ ကျွန်တော်လည်း အထွင်းခံခဲ့ရပါတယ်။ စဖတ်တုန်းက ဘာမှန်းကို မသိပါဘူး။ ဒါပေမယ့် နားလည်အောင် အတတ်နိုင်ဆုံး ကြိုးစားဖတ်ရှုပါ။ အခုနားမလည်သေးရင် ခဏထားခဲ့ပါ။ နောက်ပိုင်း concept တွေလေ့လာရင် object အကြောင်းပါလာရင် ဒီအခန်းကို ပြန်ဖတ်ရှုပါ။ တဖြည်းဖြည်းနဲ့ နားလည်သွားမှာပါ။

JavaScript Is Object-Oriented

JavaScript ဟာ Object-Based ဘာလို့ဖြစ်ရတာလဲဆိုတာကို နားလည်အောင် Object ဆိုတာဘာလဲနဲ့ သူတို့ရဲ့ အလုပ်လုပ်ဆောင်ပုံကို ပထမဆုံးအနေနဲ့ နည်းနည်း လေ့လာ ကြည့်ရအောင်။ အခြေခံအားဖြင့်တော့ Object ဆိုတာ အချက်အလက်(information) တွေကို စုစည်းထားခြင်းပဲ ဖြစ်ပါတယ်။ အဲဒီစုစည်းမှုထဲမှာ အချက်အလက်တွေကို အသုံးချကိုင်တွယ်ဖို့ method တွေလဲ တစ်ခါတည်း ပါဝင်ပါတယ်။ Object ဟာ Item တစ်ခုကို သူနဲ့ဆိုင်တဲ့ အသေးစိတ် အစိတ်အပိုင်း လေးတွေအဖြစ် ခွဲခြမ်းသတ်မှတ်ပေးပါတယ်။ အဲဒီအစိတ်အပိုင်းလေးတွေကို properties လို့လဲ ခေါ်ပါတယ်။ ပြီးတော့ ဒီအစိတ်အပိုင်းလေးတွေမှာ ဆောင်ရွက်ဖို့ တာဝန်လေးတွေလဲ ပါလာတတ်ပါတယ်။ အဲဒါကိုတော့ methods လို့ခေါ်ပါတယ်။ ဒီ Properties တွေ၊ Methods တွေကို စုပေါင်းပြီးတော့ Object လို့အမည်တပ်လိုက်တာပါပဲ။ ဒီလို Object ရဲ့ယေဘုယျ သဘော သဘာဝကြောင့် Object တစ်ခုမှာ ပါဝင်မယ့် instances တွေကို လိုတဲ့အချိန်မှာ လိုသလို ဖန်တီး အသုံးပြုနိုင်ပါတယ်။ ဥပမာ - Car ဆိုတဲ့ object တစ်ခုမှာ Toyotas၊ Fords၊ Volkswagens စသဖြင့် instances တွေအများကြီး ပါဝင်နိုင်ပါတယ်။

Working with Objects in JavaScript

JavaScript မှာ HTML document တွေအတွက်ကော၊ တစ်ခြား အသုံးဝင်တဲ့ task တွေ (ဥပမာ- mathematical calculationတွေ) အတွက်ကော built-in object တွေ ပါဝင်ပြီးသားပါ။ ဒါ့အပြင် programmer ဟာ ကိုယ်ပိုင် object တွေကိုလည်း ဖန်တီးအသုံးပြုနိုင်ပါသေးတယ်။

Built-in Objects

JavaScript မှာပါဝင်တဲ့ built-in objects တော်တော်များများဟာ Navigator Object Hierarchy ရဲ့ အစိတ်အပိုင်းတွေပါ။ Navigator Object Hierarchy မှာ ပါဝင်တဲ့ object တွေကတော့…..

- Window
- Location
- History
- Document
- Forms
- Anchors တို့ပဲ ဖြစ်ပါတရ်။

Table 1.1 Overview of the Navigator Object Hierarchy

Object	Description
window	The window object provides methods and properties for dealing with the actual Navigator window, including objects for each frame.
location	The location object provides properties and methods for working with the currently opened URL.
history	The history object provides information about the history list and enables limited interaction with the list.
document	The document object is one of the most heavily used objects in the hierarchy. It contains objects, properties, and methods for working with document elements including forms, links, anchors, and eventually, with applets.

တစ်ခြား Built-in object များ

String – string object နဲ့ စာသားတွေကို upper case ကနေ lower case ပြောင်းတာ၊ substrings တွေ ခွဲထုတ်တာနဲ့ text တွေကို ကိုင်တွယ် လုပ်ဆောင်နိုင်ပါတရ်။

Math – math object ကတော့ သချာ်ဆိုင်ရာ တွက်ချက်မှုတွေကို လုပ်ဆောင်ပေးနိုင်ပါတရ်။

Date – date object ကတော့ အချိန်၊ နေ့စွဲနဲ့ဆိုင်တဲ့ ကိစ္စတွေကို လုပ်ဆောင်ပေးနိုင်ပါတရ်။

ကိုယ်ပိုင် object များဖန်တီးခြင်း

လေယာဉ်ထုတ်လုပ်တဲ့ လုပ်ငန်းတစ်ခုက ရောင်းချမယ့် လေယာဉ်အမျိုးအစားအသီးသီးကို ကိုရ်စားပြုတဲ့ object တစ်ခုကို ဥပမာအနေနဲ့ ဖန်တီးကြည့်ရအောင်။ အဲဒီ object ထဲမှာ လေယာဉ်တွေနဲ့ ဆိုင်တဲ့ အောက်ပါ အချက်အလက် အစိတ်အပိုင်း (properties) လေးတွေ အများကြီးလဲ ထည့်သွင်းရပါမရ်။

- Model
- Price
- Normal Seating Capacity
- Normal Cargo capacity
- Maximum Speed
- Fuel capacity

Properties ဆိုတာ traditional language တွေဖြစ်တဲ့ C တို့ Pascal တို့က variable နဲ့ဆင်တူပါတရ်။ Variable ဆိုတာကတော့ တန်ဖိုး (value) တစ်ခုကိုထားသိုဖို့ အတွက်သုံးတဲ့ container ပါ။ Variable အကြောင်းကို နောက်ပိုင်းသင်ခန်းစာတွေမှာ ဆွေးနွေးပါ့မရ်။ ဒီတော့ လေယာဉ်တွေကို ကိုယ်စားပြုမယ့် object ကို airplane လို့နာမည်ပေးမရ်ဆိုရင် သူ့ရဲ့ properties တွေကို အောက်ကအတိုင်း ရည်ညွှန်းခေါ် ဆိုရပါမရ်။

- airplane.model
- airplane.price
- airplane.seating
- airplane.cargo
- airplane.maxspeed
- airplane.fuel

အပေါ်က ဥပမာကိုကြည့်ရင် JavaScript မှာ object တစ်ခုက properties တွေကို ရည်ညွှန်းချင်ရင် သုံးရမယ့် syntax ကတော့ object-name.property-name ပဲဖြစ်ပါတရ်။

Methods

ဒီအချက်အလက်တွေကို အသုံးချဖို့ နည်းလမ်းမရှိဘဲနဲ့ အချက်အလက်တွေ အတိုင်းကြီးပဲ ဆိုရင် ဘရ်အဖိုး သိပ်တန်မလဲနော်? အထက်က ဥပမာမှာ လေယာဉ်တစ်စီးရဲ့ အကြောင်း ဖော်ပြချက်ကို print out လုပ်ပြချင်တရ်။ ဒါမှမဟုတ် လောင်စာပေါ်မူတည်ပြီး လေယာဉ်ဟာ ဘရ်လောက်ဝေးဝေး သွားနိုင်တရ်ဆိုတာကို တွက်ထုတ်ပြချင်တရ်လို့ပဲ ဆိုကြပါစို့။ Object-Oriented အသုံးအနှုန်းနဲ့ပြောရရင် ဒီလို ဆောင်ရွက်ချက်တွေကို method လို့ခေါ် ပါတရ်။ Properties တုန်းကလိုပဲ method တွေကို ရည်ညွှန်းခေါ် ဆိုချင်ရင်…

- airplane.description( )
- airplane.distance( )

Object အတွင်းက Object

Object တစ်ခုအတွင်းမှာ properties တွေ၊ method တွေပါဝင်သလိုပဲ နောက် object ခွဲတွေလည်း ပါဝင်နိုင်ပါတရ်။

- airplane.record.number_in_use
- airplane.record.crashes

ဒါကတော့ javascript အားအသုံးပြုခြင်းရဲ့ အပိုင်း(၁)လေးပါ။ အားလုံးက အခြေခံလေးတွေ ဖြစ်လို့ စတင်လေ့လာသူများ အတွက်သာ ရည်ရွယ်ပါတရ်။ ကျနော့် မှားယွင်းတင်ပြမှုများ ရှိရင်လဲ နားလည်ပေးကြပေါ့ဗျာ။ ကျနော် ရည်ရွယ်ချက်ကတော့ အခြေခံလေးတွေကို မြန်မာလို ဖတ်ရှုခြင်းအားဖြင့် လျှင်မြန်စွာ နားလည် သွားနိုင်ပြီး ဒီထပ် မြင့်သော တခြားသော နယ်ပယ်များကို ကူးသွားနိုင်အောင်ပါ။ မိတ်ဆွေအပေါင်း လေ့လာခြင်းဖြင့် ကျေနပ်နိုင်ကြပါစေ။

Tay Zar Lin
Koyinmaung007@gmail.com
Programmingknowledge.blogspot.com