**برمجة إضافات جوجل كروم**



متصفح جوجل كروم أصبح من أفضل المتصفحات على الإطلاق لما يوفره من أدوات و إضافات تسهل عملية التصفح و تجعل من تطوير الويب أمرا ممتعا بكل المقاييس.

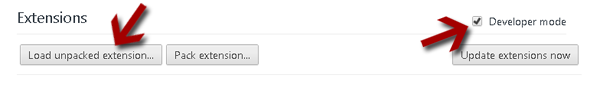
ﻻ شك في أنك جربت إضافات متصفح جوجل كروم من قبل،و مهما كانت هته الإضافة،فلا بد أنك تساءلت كيف تمت عملية برمجة و تطوير هته الإضافات،هل هذه العملية صعبة؟ما هي لغات البرمجة التي علي تعلمها حتى استطيع برمجة إضافات جوجل كروم الخاصة بك؟

أبشر،إن كنت مطور مواقع ويب فلن تحتاج لتعلم أية لغة برمجة أخرى،فإضافات متصفح جوجل كروم تتم برمجتها عن طريق لغات الويب HTML و CSS و Java Script الثلاثة:

ﻻ غير،و لكن عليك اتقان هته اللغات بشكل جيد جدا.

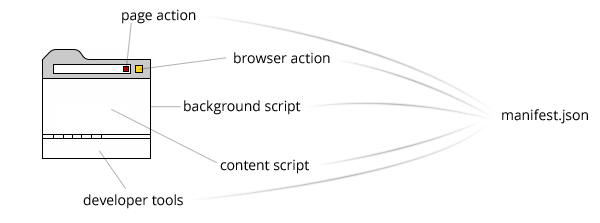
قبل أن نبدأ،دعونا نتعرف على بنية إضافات جوجل كروم أوﻻ ثم ننتقل بعدها إلى كيف تتم البرمجة.عليك أن تقوم اﻵن يتفعيل من صفحة الإضافات:افتح المتصفح و اكتب في العنوان: Developer Mode

[chrome://extensions](http://shorte.st/strackable/38caccdcc22ee70e9f0a428fa4976b8c/www.almaggeek.com/1/1/chrome:/extensions/)



ثم نختار Load unpacked extension و لكن هذا بعد أن ننهي عملية البرمجة حتى ﻻ نضطر إلى رفعها إلى متجر إضافات جوجل كروم في كل مرة نقوم بالتعديل على الإضافة.

الآن لنتعرف على بنية إضافات جوجل كروم:دعونا نرى الصورة الموالية



تتكون إضافات جوجل كروم من جزئين رئيسيين:الجزء البرمجي و واجهة المستخدم،و سنبدأ بالجزء البرمجي،و الذي بدوره ينقسم إلى ثلاثة أجزاء رئيسية:

manifest.json :و هو ملف أساسي ﻷي إضافة،يحوي هذا الملف على معلومات مهمة عن الإضافة،كإسم الإضافة،رقم الإصدار و وصف الإضافة و يمكن أن يحوي العديد من المعلومات الأخرى،يمكنك الإطلاع عليها <http://goo.gl/ZIOd3F>  [من هنا](https://developer.chrome.com/extensions/manifest.html).دعونا نأخذ مثاﻻ بسيطا

{

    "name": "BrowserActionExtension",

    "version": "0.0.1",

    "manifest\_version": 2,

    "browser\_action": {

        "default\_title": "That's the tool tip",

        "default\_popup": "popup.html"

    }

}

سأشرح كل سطر على حدى:  
name :و هو اسم الإضافة  
version : إصدار الإضافة  
manifest\_version: هي إصدار "محرك" المتصفح،و ضعه دائما 2  
browser\_action:هي تعريف ببعض الخصائص  
default\_title: التسمية التي تظهر في المتصفح  
default\_popup: و تمثل ملف واجهة الإضافة

Background Files: أو الملفات الأساسية و يمكن أن نعتبرها القلب النابض للإضافة و التي تحوي الأكواد البرمجية الأساسية لها،و تنقسم إلى قسمين:

persistent: أو الأكواد الأساسية التي يتم تشغيلها بمجرد فتح المتصفح و تبقى في حالة العمل ما دام المتصفح شغاﻻ

event:

أو اكواد اﻷحداث و يتم تشغيلها عند حدث معين كأن يطلب المستخدم القيام بشيء معين من الإضافة.يتم تعريف كلا النوعين في ملف manifest.json:

كالتالي

"background": {  
    "scripts": ["background.js"],  
    "persistent": false/true  
}

Script :

تمثل مكان تواجد الملف الذي يحوي الأكواد أما Persistent فهي خيار منطقي إما مفعل أو غير مفعل،سنفصل في هته الجزئية في الدروس المقبلة بحول الله.

Content Script: و نعني بها في أي مكان سيتم تشغيل الإضافة،فهنالك إضافات ﻻ تعمل إﻻ في موقع واحد مثل إضافة موقع ميجا الخاص بالرفع فهي ﻻ تعمل إﻻ عندما تتصفح الموقع أو تقوم أنت بفتحها يدويا و هنالك إضافات تعمل مع جميع المواقع دون استثناء،يتم manifest.json: تعريف ذلك أيضا في ملف

كالتالي

"content\_scripts": [  
    {  
        "matches": ["http://\*/\*", "https://\*/\*"],  
        "js": ["content.js"]  
    }  
]

Browser\_action خاصية

،فقد ذكرت أنها تعريف لبعض الخصائص،نعم فهي تستخدم للعديد من اﻷمور منها تعيين أيقونة للإضافة،شاهد المثال التالي



"browser\_action": {  
    "default\_icon": {  
        "19": "icons/19x19.png",  
        "38": "icons/38x38.png"  
    },  
    "default\_title": "That's the tool tip",  
    "default\_popup": "popup.html"  
}

بالطبع يتم تعريفها في ملف

manifest.json

،أعتقد أن كل شيء واضح و ﻻ يستدعي الشرح المستفيض له،الآن للنتقل لجزئية أخرى و هي التسمية التي

تظهر مع أيقونة الإضافة و أقصد هنا الكتابة yeah فيتم إضافتها عن طريق السطر التالي في نفس الملف:

chrome.browserAction.setBadgeText({text: "yeah"});

هنالك خاصية أخرى يمكن إستخدامها في تطوير تطبيقات متصفح كروم و هي page\_action

كمثال عليها



و يتم تعريفها في نفس الملف السابق كالتالي:

"page\_action": {  
    "default\_icon": {  
        "19": "images/icon19.png",  
        "38": "images/icon38.png"  
    },  
    "default\_title": "Google Mail",  
    "default\_popup": "popup.html"  
}

 بصراحة هنالك العديد من الإضافات،و ﻻ يمكنني بأي حال من الأحوال ذكرها كلها،و بالتالي إن كنت تريد معرفة المزيد من الخصائص فالأجدر بك زيارة موقع مطوري جوجل http://goo.gl/kA94MO  [من هنا](https://developer.chrome.com/extensions/devguide.html).

الان سنتعرف على أشياء جديدة مهمة جدا تتعلق بهذا المتصفح

نكمل سويا ما بدأناه من مسيرة التعرف على بنية متصفح جوجل كروم،حتى نستطيع فيما بعد برمجة و تصميم إضافات خاصة بنا له...اليوم سنتعرف على أشياء جديدة مهمة جدا تتعلق بهذا المتصفح.

أدوات المطورين Developer Tools:

تمكنك هته الخاصية من إضافة ألسنة جديدة لتطبيقك و بالتالي يمكنك جعل اﻹضافة تعمل ﻷكثر من مرة،حتى تستطيع استخدام هته الخاصية عليك إنشاء صفحة ويب جديدة و بعدها تطلبها في ملف manifest.json

كاﻵتي:

"devtools\_page": "devtools.html"

في داخل الملف نقوم بتضمين ملف الأكواد البرمجية الخاصة بالجافاسكريبت كالتالي

كما تلاحظون علينا انشاء ملف جديد بإسم

devtools.js

يمكنكم تسميته كما تشاؤون و لكن عليكم تغيير الإسم عند القيام بتضمين:

الملف.في ملف الجافاسكريبت ضع الكود التالي

chrome.devtools.panels.create(  
    "TheNameOfYourExtension",  
    "img/icon16.png",  
    "index.html",  
    function() {  
  
    }  
);

أعتقد أن الكود بسيط و مفهوم و ﻻ يحتاج لشرح دقيق،فلا شيء مبهم فيه...لننتقل إلى نقطة أخرى...:

اOmnibox:الـ

متصفح جوجل كروم يتيح لنا إصدار أوامر كتابية له لتنفيذها،هذه الأوامر هي في اﻷصل كلمات مفتاحية تم وضعها من قبل مبرمجي الإضافات.

manifest.json يتم تعريف الكلمات المفتاحية في ملف

و تتم العملية كالأتي:

"omnibox": { "keyword" : "yeah" }

omnibox تلاحظون بأنه تم تعريف

Background- على أنه مصفوفة(قاموس).يتم تعريف الكلمات المفتاحية في ملف الخلفية

تتذكرونه- كاﻵتي:

chrome.omnibox.onInputChanged.addListener(function(text, suggest) {  
    suggest([  
      {content: text + " one", description: "the first one"},  
      {content: text + " number two", description: "the second entry"}  
    ]);  
});  
chrome.omnibox.onInputEntered.addListener(function(text) {  
    alert('You just typed "' + text + '"');  
});

لو قمنا بكتابة

في المتصفح لتحصلنا على التالي: yeah

: C:\Users\rtfgi$koli$krtfgedu\Desktop\omnibox.png

كيف حالكم مع متصفح جوجل كروم لحد الان ؟

بطبيعة الحال يوفر جوجل كروم العديد من الوظائف عن طريق الـ

API

Application Programming Interface ،لمن ﻻ يعرفها فهي اختصار لجملة

،و هي تخص الوظائف التي يوفرها جوجل كروم أو

يستخدمها في عمله و يتيح للمطور استخدامها و الإستفادة منها في تطبيقاته.

كما ذكرت قبل قليل،فإن كروم يوفر العديد من الوظائف و ﻻ يمكن بأي حال من الأحوال تغطيتها جميعا و لكن عليك الإطلاع على التوثيق الرسمي لها حتى تستطيع فهم آلية العمل مع معرفة كيفية الإستخدام.

<http://goo.gl/SIlN3l> [من هنا](http://shorte.st/strackable/38caccdcc22ee70e9f0a428fa4976b8c/www.almaggeek.com/1/1/https:/developer.chrome.com/extensions/api_index.html). يمكنك زيارة التوثيق الرسمي لها

ننتقل إلى الجزء الثاني و هو الرسائل،تسمى رسائل و لكنها ليست كذلك.هي في الحقيقة أداة مراقبة ﻷداء الإضافة أو القيام بعمل ما استنادا لنتيجة معينة و هي نوعان:

**One-Time Request:**

سأعطيكم مثاﻻ بسيطا حتى تتضح الأمور،لو اردنا الإستفسار أو اخذ معلومات عن صفحة معينة،و نريد اخذ معلومات تخص

 الخاص بالصفحة...سنقوم بكتابة الكود التالي: DOM الـ

chrome.extension.onMessage.addListener(function(request, sender, sendResponse) {  
    switch(request.type) {  
        case "dom-loaded":  
            alert(request.data.myProperty);  
        break;  
    }  
    return true;  
});

background يتم كتابة هذا الكود في صفحة الـ

(اعتقد أنكم تتذكرونها) و في ملف

content

سنضع الكود التالي ﻻستقبال المعلومات

window.addEventListener("load", function() {  
    chrome.extension.sendMessage({  
        type: "dom-loaded",  
        data: {  
            myProperty: "value"  
        }  
    });  
}, true);

 **Long-lived Connection:**

content يستخدم هذا النوع ﻹنشاء اتصال دائم،و ذلك عن طريق وضع الكود التالي في صفحة الـ

var port = chrome.runtime.connect({name: "my-channel"});  
port.postMessage({myProperty: "value"});  
port.onMessage.addListener(function(msg) {  
    // do some stuff here  
});

background و في صفحة الـ

نضع الكود التالي

chrome.runtime.onConnect.addListener(function(port) {  
    if(port.name == "my-channel"){  
        port.onMessage.addListener(function(msg) {  
            // do some stuff here  
        });  
    }  
});:

 لم يتم شرحها بالتفصيل في الجانب التطبيقي ﻷنها أمور برمجية لن تستطيع فهمها إن لم تكن تمتلك خبرة بسيطة مع لغة الجافاسكريبت

[www.intrprof.com](http://www.intrprof.com)