**مقدمة في النصوص Strings**

**مقدمة في مفهوم النصوص Strings**

تعتبر النصوص Strings من المواضيع المهمة في البرمجة، وحتى نجعل الأمور أكثر وضوحاً، يمكننا القول بأن النص String هو عبارة عن سلسلة مكونة من مجموعة من الأحرف Characters والتي يتم وضعها بين علامات التنصيص المفردة Single-Quote علامات التنصيص المزدوجة Double-Quote أو علامتي Backticks.

**ماذا نعني بالنوع String ؟**

ببساطة يمكننا القول بأن النص String: هو عبارة عن سلسلة أو مجموعة من الحروف Characters التي تكون مُحاطة بعلامات تنصيص مفردة Single-Quote أو مزدوجة Double-Quote أو أن تكون مُحاطة بعلامات Backticks. عندما نقول حروف هنا فهذا الأمر يشمل الأرقام، الرموز، والمسافات وما إلى ذلك، وحتى نجعل الأمور أكثر وضوحًا نلاحظ المثال التالي:

console.log('Welcome to JavaScript');

ببساطة، قمنا بطباعة رسالة ترحيبية بسيطة، ونلاحظ أن الرسالة مُحاطة بعلامتي تنصيص، وهنا يمكننا القول بأن هذه الرسالة هي عبارة عن String، وفي حال قمنا بتشغيل البرنامج، فستكون المُخرجات على النحو التالي:

Welcome to JavaScript

نلاحظ هنا أن المخرجات لاتشمل علامات التنصيص.

**طرق كتابة String**

يمكننا كتابة المثال السابق بأكثر من طريقة، حيث تتيح لنا JavaScript كتابة النص السابق بثلاث طرق مختلفة منها الطريقة التي استخدمناها من قبل، ولتوضيح الفكرة نلاحظ المثال التالي:

console.log('Welcome to JavaScript');

console.log("Welcome to JavaScript");

console.log(`Welcome to JavaScript`);

نلاحظ هنا أننا قمنا باستخدام علامات مُختلفة لتمثيل النص في كل مرة، ففي المرة الأولى استخدمنا علامات التنصيص المفردة Single-Quote وفي الثانية استخدمنا علامات التنصيص المزدوجة Double-Quote وفي الثالثة استخدمنا علامات Backticks وهو مايُعرف بمفهوم Templete String وسنتعرف على هذا النوع بشكل مفصل في الدرس القادم.

Welcome to JavaScript

Welcome to JavaScript

Welcome to JavaScript

نلاحظ أن المخرجات هي نفسها في هذه الحالة، سنتعرف فيما بعد على بعض فوائد استخدام هذه الطرق المختلفة. باختصار فكر في النص String على أنه مجموعة من الرموز والتي تأتي كوحدة واحدة محاطة بعلامات تنصيص مفردة أو مزدوجة أو backticks.

المراجع

* https://javascript.info/string
* https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/String

**مقدمة في مفهوم Template Strings**

سنتطرق في هذا الدرس إلى مفهوم Template Strings والذي من خلاله يمكنك كتابة نص متعدد الأسطر، بالإضافة إلى إمكانية إدراج متغيرات وعمليات مختلفة داخل نص معين.

**مفهوم Template String**

توفر لك لغة JavaScript طريقة تُسمى Template Strings والتي تمكنك من كتابة نص مُتعدد الأسطر، بالإضافة إلى إمكانية كتابة نص يحتوي على مُتغيرات وأوامر برمجية مُختلفة، لإنشاء Template Strings سنستخدم علامة Backtick ولتوضيح الفكرة نلاحظ المثال التالي:

let message = `Welcome to JavaScript`;

قمنا بتعريف متغير باسم message يحتوي على نص يمثل رسالة ترحيبية، نلاحظ هنا أن النص مُحاط بعلامتي Backticks وليس علامتي تنصيص مفردة Single-Quote، وبهذه الطريقة نكون قد قمنا بتعريف نص من النوع Template Strings.

**كتابة نص متعدد الأسطر**

لكتابة نص مُتعدد الأسطر يمكنك استخدام Template Strings بالشكل الموضح في المثال التالي:

let message = `Welcome to

JavaScript`;

console.log(message);

نلاحظ أن قيمة المتغير message قد كُتبت على سطرين وعند طباعة المتغير message ستكون المُخرجات على النحو التالي:

Welcome to

JavaScript

الآن نلاحظ أن النتيجة مُطابقة لقيمة المتغيّر message.

ملاحظة: يجب الأخذ بعين الاعتبار أن النص مُتعدد الأسطر حساس للمسافات، أي أنه سيأخذ الشكل الذي كُتب عليه.

**إدراج Expression داخل Template Strings**

يمكننا إدراج تعبير برمجي Expression داخل Template Strings باستخدام العلامات ${}، ولتوضيح الفكرة نلاحظ المثال التالي:

let price = 5.75;

let message = `The price is ${price}`;

console.log(message);

نلاحظ كيف قمنا بإدراج المتغيّر price داخل القيمة النصية للمتغيّر message، وعليه ستكون المُخرجات لأمر الطباعة الموضح في السطر الثالث على النحو التالي:

The price is 5.75

لاحظ الآن كيف تم استبدال المتغير price بقيمته في النتيجة النهائية.

لايقتصر الأمر على إدراج المتغيرات، فبإمكانك أيضًا إدراج عمليات مُختلفة، ولتوضيح الفكرة نلاحظ المثال التالي:

console.log(`The sum is ${1 + 2}`);

نلاحظ أننا قمنا بإدراج عملية حسابية داخل النص، وبإمكانك استخدام عمليات مُختلفة سواءً باستخدام القيم المباشرة أو المتغيرات، وستكون المُخرجات في هذهِ الحالة على النحو التالي:

The sum is 3

تم تنفيذ العمليّة الحسابيّة أولاً، وبعد ذلك تم إدراج الناتج ودمجه مع النص النهائي كما هو ظاهر في المُخرجات.   
تعرفنا في هذا الدرس على مفهوم ومميزات Template Stringsوكيفية استخدامها وتوظيفها لكتابة نص مُتعدد الأسطر بالإضافة إلى استخدامها لإدراج عمليات مُختلفة داخل النصوص.   
المراجع

* https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Template\_literals
* JavaScript: The Definitive Guide, 7th Edition by David Flanagan
* Learning JavaScript, 3rd Edition by Ethan Brown

**مقدمة في خاصية length**

يعتبر حساب عدد الأحرف في نص معين من العمليات التي يتم استخدامها في كثير من البرامج لخدمة أغراض مُختلفة، فمثلًا قد تحتاج إلى التحقق من طول كلمة السر أثناء تسجيل مستخدم جديد في برنامجك، وهنا يأتي دور خاصيّة lengthفي هذا الدرس سوف نتعرف على خاصيّة lengthوكيفية كتابتها واستخدامها. استخدام length لمعرفة عدد الأحرف يمكن حساب عدد الأحرف في نص معين عن طريق استخدام خاصيّة length ولتوضيح الفكرة نلاحظ المثال التالي:

let name = 'Nasser';

console.log(name.length);

في السطر الأول قمنا بتعريف متغير باسم name وقيمته Nasserوفي السطر الثاني قمنا بحساب عدد الأحرف لقيمة المتغير nameمن خلال استخدام خاصيّة lengthومن ثم قمنا بطباعة النتيجة وفي هذه الحالة ستكون المخرجات على النحو التالي:

6

نلاحظ أن النتيجة 6 تمثل عدد أحرف قيمة المتغير name أي القيمة Nasser، وهذا باختصار هو دور خاصية length. في حال كان لدينا نص فارغ وقمنا باستخدام خاصية length فإن النتيجة ستكون صفر.

**استخدام length مع النص بشكل مباشر**

يمكن استخدام خاصية length بشكل مباشر مع نص معين، ولتوضيح الفكرة نلاحظ المثال التالي:

console.log('Saudi'.length);

نلاحظ أننا قمنا باستخدام خاصية lengthمباشرة مع النص Saudi وستكون المخرجات على النحو التالي:

5

كما نلاحظ حصلنا على النتيجة 5 وهذا لأن كلمة Saudi تتكون من5 أحرف.

للتنبيه، يتم احتساب المسافات والرموز عند حساب طول النص.

**مثال على استخدام length**

من الاستخدامات الشائعة لخاصية length استخدامها في برنامج يقوم بحساب طول كلمة السر لمستخدم جديد في حال اشتراط طول معين لقبول كلمة السر ولتوضيح الفكرة نلاحظ المثال التالي:

let password = '\_@2#bcE?';

if(password.length >= 8){

console.log("The password is accepted") ;

}

كما نلاحظ في السطر الأول قمنا بحفظ كلمة السر داخل المتغير password بعد ذلك قمنا بوضع الشرط الخاص لقبول كلمة السر، وهو أن تتكون 8 أحرف أو أكثر، ويتم طباعة الجملة The password is accepted في حال تحقق الشرط، وعليه ستكون المخرجات على النحو التالي:

The password is accepted

نلاحظ أن الشرط تحقق وتمت طباعة الجملة The password is accepted وذلك لأن طول كلمة السر يساوي 8. تستخدم if للتحقق من صحة شرط معين، لذلك قمنا باستخدامها للتحقق من عدد حروف كلمة المرور وسنتعرف على جملة if بشكل مفصل في الدروس القادمة بإذن الله. تعلمنا في هذا الدرس كيفية حساب ومعرفة عدد الأحرف في نص معين من خلال استخدام الخاصية length والطرق المختلفة لكتابتها واستخدامها.   
المراجع

* Learning JavaScript, 3rd Edition
* <https://developer.mozilla.org/en-US/>

**الوصول الى العناصر باستخدام**

تتيح لك لغة JavaScript ‬إمكانية الوصول الى عنصر او حرف معين داخل النص، وسنتعرف في هذا الدرس على كيفيتها من خلال الاقواس المربعة [] او ما يسمى بمفهوم Bracket Notation .

**كيفية الوصول إلى عنصر معين داخل النص**

يمكن الوصول إلى عنصر معين داخل النص عن طريق استخدام الأقواس المربعة [] أو Bracket Notation وكتابة ‬رقم خانة ‬العنصر المطلوب بينهما ولتوضيح ذلك ‬لاحظ معي المثال التالي:

let language = 'JavaScript';

console.log(language[2]);

قمنا في السطر الثاني من المثال السابق بتعريف متغير language يحتوي على القيمة JavaScript ويتم ترقيم الحروف في النص في لغة JavaScript من اليسار إلى اليمين بحيث يبدأ تعداد أول حرف من صفر، أي أن الحرف الأول J ستكون قيمة خانته صفر ورقم خانة الحرف الثاني a هي 1 وهكذا، وعند طباعة الخانة رقم 2 للمتغير language ستكون المخرجات على النحو التالي:

v

نلاحظ أن النتيجة هي v وذلك لأن قيمة الخانة 2 في المتغير language هي الحرف v.

يُعرف رقم الخانة والذي نضعه بين الأقواس المربعة بمفهوم Index.

**تحديث عنصر داخل النص**

تستخدم Bracket notation مع النصوص للقراءة فقط، أي لايمكنك تحديث قيمة عنصر معين داخل النص باستخدام Bracket Notation ولتوضيح هذهِ الفكرة نلاحظ المثال التالي:

let name = 'Mishal';

name[0] = 'R';

console.log(name);

في المثال السابق قمنا بتعريف متغير باسم name وقيمته Mishalوفي السطر الثاني قمنا بمحاولة لتعديل الخانة الأولى للمتغير name إلى الحرف R، وعند طباعة المتغير name ستكون المخرجات على النحو التالي:

Mishal

نلاحظ أن قيمة الخانة رقم 0 للمتغير name لم تتغير إلىR بل بقيت كما هي على قيمتها الأساسية، أي M.   
تعرفنا في هذا الدرس على كيفية الوصول إلى عنصر معين داخل النص وتنفيذ عمليات مُختلفة عليه مثل الطباعة، كما تعلمنا أن Bracket notation خاصة بالقراءة ولا تقوم بتحديث العناصر أو تغييرها.

المراجع

* JavaScript Pocket Reference, 3rd Edition
* https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/String

**مقدمة في دمج النصوص String Concatenation**

تتيح لك لغة JavaScript عدة طرق لدمج النصوص مع بعضها البعض، وأيضًا دمج النصوص مع المتغيرات، وهو ما يعرف بمفهوم String Concatenation.

**دمج نصين مع بعضهما البعض**

يمكنك دمج نصين باستخدام علامة الجمع + ، ولتوضيح ذلك سنقوم بكتابة نصين بداخل أمر الطباعة وسنقوم بدمجهما باستخدام علامة + لتكون النتيجة النهائية هي عبارة عن نص واحد، يوضح السطر التالي هذا الأمر:

console.log ( 'Welcome to ' + 'JavaScript');

قمنا في المثال السابق بدمج النص Welcome to مع النص JavaScript وذلك باستخدام العلامة +، وستكون المخرجات على النحو التالي:

Welcome to JavaScript

نلاحظ أن نتيجة المُخرجات توضح عملية الدمج التي تمت بين النصين ليكونا نصًا واحدًا، بالإضافة إلى ذلك نلاحظ أننا وضعنا مسافة واحدة في آخر النص الأول، وتحديدًا بعد كلمة toوذلك لتلافي إلتصاق النصين ببعضهما البعض بشكل مباشر ودون وجود مسافة.

**دمج نص مع متغير**

بنفس الأسلوب يمكننا دمج نص مع متغير معين ليكون الناتج النهائي هو النص مضافًا له قيمة ذلك المُتغير ولتوضيح الفكرة، نلاحظ المثال التالي:

let language = 'JavaScript';

console.log('Welcome to ' + language);

نلاحظ كيف قمنا بدمج النص Welcome to مع قيمة المتغير language في السطر الثاني، ونلاحظ أيضًا وجود المسافة بعد كلمة to وعليه ستكون المُخرجات على النحو التالي:

Welcome to JavaScript

توضح المُخرجات النتيجة النهائية بعد عملية الدمج والتي تظهر فيها رسالة الترحيب بشكلها الكامل.

**الدمج بأسلوب مُختصر**

هناك أسلوب مُختصر لعملية الدمج وذلك من خلال استخدام += ولتوضيح الفكرة نلاحظ المثال التالي:

let title = 'Mr.';

title += 'Ali'; // title = title + 'Ali';

console.log(title);

في هذا المثال قمنا بتعريف متغير باسم title وفي السطر الثاني قمنا بدمج القيمة Ali إلى القيمة السابقة للمتغير title بطريقة مُختصرة من خلال +=، وعليه ستكون المُخرجات على النحو التالي:

Mr.Ali

نلاحظ كيف أصبحت النتيجة النهائية بعد الدمج، ولجعل الأمور أكثر وضوحًا يمكننا القول أن استخدام += في السطر الثاني من المثال السابق هو طريقة مُختصرة للسطر التالي:

title = title + 'Ali';

بمعنى قم بدمج القيمة الحالية للمتغير title مع القيمة Ali ومن ثم قم بتخزين النتيجة النهائية في المتغير title. تعرفنا في هذا الدرس على كيفيه دمج النصوص مع بعضها البعض بالإضافة إلى دمج النصوص مع المتغيرات، كما تعرفنا على كيفية دمج قيمتين بطريقة مُختصرة.

المراجع

* JavaScript: The Definitive Guide, 7th Edition
* https://javascript.info/operators#string-concatenation-with-binary

**مقدمة في مفهوم Escape Character**

سنتعرف في هذا الدرس على Escape Characters وهي عبارة عن رموز تستخدم مع النصوص ولكل منها تأثير معين، فمنها مايقوم بإضافة سطر ومنها مايقوم بإدراج رمز معين وما إلى ذلك، هناك مجموعة كبيرة من Escape Characters وسنتناول في هذا الدرس عدد من هذه الرموز.

**إضافة Tab داخل النص**

يقوم \t بإضافة عدة مسافات بين العناصر التي وضع بينها ولتوضيح الأمر سنقوم الآن بكتابة نص مكون من 5 أحرف وسنقوم باستخدام \t لوضع مسافة بين كل حرف والآخر على النحو التالي:

console.log('A\tB\tC\tD\tE');

في المثال السابق قمنا بكتابة أمر طباعة يطبع لنا الأحرف من A إلى E، نلاحظ كيف قمنا بكتابة \t بين كل حرف والآخر، بعد تنفيذ الأمر السابق ستكون المخرجات على النحو التالي:

A B C D E

كما هو موضح في المُخرجات، يوجد الآن عدد من المسافات بين الأحرف وهذا هو الدور الذي يقوم به \t.

**إضافة New Line داخل النص**

يقوم \n بوضع سطر جديد بعده ولتوضيح الأمر سنقوم الآن بكتابة نص مكون من 5 أحرف وسنستخدم \n لجعل كل حرف من تلك الأحرف على سطر جديد كما هو موضح في المثال التالي:

console.log('A\nB\nC\nD\nE');

في المثال السابق قمنا بكتابة أمر طباعة يطبع لنا الأحرف A و B و C و D و E، نلاحظ كيف قمنا بكتابة \n بين كل حرف والآخر، بعد تنفيذ السطر السابق ستكون المخرجات على النحو التالي:

A

B

C

D

E

كما نلاحظ تمت طباعة كل حرف على سطر جديد.

**إضافة علامة التنصيص المفردة**

يقوم \' بإدراج علامة التنصيص المفردة داخل النص. لتوضيح الأمر سنقوم الآن بكتابة نص يحتوي على علامة التنصيص المفردة كما هو موضح في السطر التالي:

console.log('It\'s such an interesting language');

في السطر السابق قمنا بكتابة نص باستخدام علامتي التنصيص المفردة، وكان النص عبارة عن جملة تحتوي على علامة التنصيص المفردة، نلاحظ كيف قمنا بكتابة \' داخل النص بعد تنفيذ السطر السابق ستكون المخرجات على النحو التالي:

It's such an interesting language

لو لم نقم باستخدام الحرف \' واكتفينا بكتابة علامة التنصيص مباشرة داخل النص، فإننا سنحصل على رسالة خطأ لأن البرنامج سيُنهي النص بمجرد وصوله لعلامة التنصيص الثانية.

**إضافة علامة التنصيص المزدوجة**

يقوم \" بإدراج علامة التنصيص المزدوجة داخل النص ولتوضيح الأمر سنقوم الآن بكتابة نص ونضع بداخله علامتي التنصيص المزدوجة حول كلمة JavaScript كما هو موضح في السطر التالي:

console.log("Welcome to \"JavaScript\"");

في السطر السابق قمنا بكتابة جملة تحتوي على علامتي تنصيص كما ذكرنا سابقًا، نلاحظ كيف قمنا بكتابة \" داخل النص وأننا قمنا بكتابة JavaScript كذلك بين علاماتي تنصيص مزدوجة بعد تنفيذ السطر السابق ستكون المخرجات على النحو التالي:

Welcome to "JavaScript"

كما هو الحال في المثال السابق، لو لم نقم باستخدام \" واكتفينا بكتابة علامتي التنصيص مباشرة داخل النص، فإننا سنحصل على رسالة خطأ لأن البرنامج سيُنهي النص بمجرد وصوله لعلامة التنصيص الثانية.

**إضافة علامة Backslash**

يقوم \\ بإدراج علامة Backslash داخل النص ولتوضيح الأمر سنقوم الآن بكتابة نص ونضع \\ داخله، كما هو موضح في السطر التالي:

console.log('The \\ character is called backslash.');

في السطر السابق قمنا بكتابة جملة تحتوي على الرمز \\، كما سبق وذكرنا نلاحظ كيف قمنا بكتابته داخل النص، بعد تنفيذ السطر السابق ستكون المخرجات على النحو التالي:

The \ character is called backslash.

لن تظهر Backslash في حال تمت كتابتها مفردة.

يمكننا الاستفادة من Escape Character واستخدامها لإتمام بعض المهام كإضافة سطر جديد وإضافة المسافات وإدراج علامات التنصيص والرموز وما إلى ذلك.

المراجع

* Learning JavaScript, 3rd Edition .(2016). By Ethan Brown.
* https://javascript.info/string#special-characters