Je veux réviser le concept de DOM en javascript , réalise un qcm de 8 question avec un niveau de difficulté moyen, me permettant d'évaluer mon niveau. Pour chaque question tu attends que je réponde avant de me poser la suivante. A la fin, tu me donnes des pistes pour m'améliorer selon mes résultats.

Sent message. Je veux réviser le concept de DOM en javascript , réalise un qcm de 8 question avec un niveau de difficulté moyen, me permettant d'évaluer mon niveau. Pour chaque question tu attends que je réponde avant de me poser la suivante. A la fin, tu me donnes des pistes pour m'améliorer selon mes résultats.

D’accord, voici un qcm sur le DOM en javascript. Bonne chance!

**Question 1:** Qu’est-ce que le DOM?

* A) Un objet qui représente la structure d’une page web
* B) Un langage de programmation basé sur le HTML
* C) Un outil de développement intégré à la plupart des navigateurs
* D) Un acronyme pour Document Object Model

Répondez avec la lettre correspondant à la bonne réponse.

Received message. D'accord, voici un qcm sur le DOM en javascript. Bonne chance! \*\*Question 1:\*\* Qu'est-ce que le DOM? - A) Un objet qui représente la structure d'une page web - B) Un langage de programmation basé sur le HTML - C) Un outil de développement intégré à la plupart des navigateurs - D) Un acronyme pour Document Object Model Répondez avec la lettre correspondant à la bonne réponse.

2of30

D

Sent message. D

Bonne réponse! Le DOM est l’acronyme pour Document Object Model, qui est un objet qui représente la structure d’une page web et permet de la manipuler avec du code.

**Question 2:** Comment accéder à un élément du DOM par son identifiant?

* A) document.getElementById(id)
* B) document.getElement(id)
* C) document.id
* D) document.querySelector(id)

Répondez avec la lettre correspondant à la bonne réponse.

Received message. Bonne réponse! Le DOM est l'acronyme pour Document Object Model, qui est un objet qui représente la structure d'une page web et permet de la manipuler avec du code. \*\*Question 2:\*\* Comment accéder à un élément du DOM par son identifiant? - A) document.getElementById(id) - B) document.getElement(id) - C) document.id - D) document.querySelector(id) Répondez avec la lettre correspondant à la bonne réponse.

3of30

A

Sent message. A

Bonne réponse! La méthode document.getElementById(id) permet de sélectionner un élément du DOM par son attribut id, qui est unique.

**Question 3:** Comment modifier le contenu texte d’un élément du DOM?

* A) element.textContent = “nouveau texte”
* B) element.innerHTML = “nouveau texte”
* C) element.text = “nouveau texte”
* D) element.content = “nouveau texte”

Répondez avec la lettre correspondant à la bonne réponse.

Received message. Bonne réponse! La méthode document.getElementById(id) permet de sélectionner un élément du DOM par son attribut id, qui est unique. \*\*Question 3:\*\* Comment modifier le contenu texte d'un élément du DOM? - A) element.textContent = "nouveau texte" - B) element.innerHTML = "nouveau texte" - C) element.text = "nouveau texte" - D) element.content = "nouveau texte" Répondez avec la lettre correspondant à la bonne réponse.

4of30

A

Sent message. A

Bonne réponse! La propriété element.textContent permet de modifier le contenu texte d’un élément du DOM, sans interpréter le HTML.

**Question 4:** Comment ajouter un élément enfant à un élément parent du DOM?

* A) parent.appendChild(child)
* B) parent.append(child)
* C) parent.add(child)
* D) parent.insert(child)

Répondez avec la lettre correspondant à la bonne réponse.

Received message. Bonne réponse! La propriété element.textContent permet de modifier le contenu texte d'un élément du DOM, sans interpréter le HTML. \*\*Question 4:\*\* Comment ajouter un élément enfant à un élément parent du DOM? - A) parent.appendChild(child) - B) parent.append(child) - C) parent.add(child) - D) parent.insert(child) Répondez avec la lettre correspondant à la bonne réponse.

5of30

A

Sent message. A

Bonne réponse! La méthode parent.appendChild(child) permet d’ajouter un élément enfant à la fin d’un élément parent du DOM.

**Question 5:** Comment supprimer un élément du DOM?

* A) element.remove()
* B) element.delete()
* C) element.parentNode.removeChild(element)
* D) element.parentNode.deleteChild(element)

Répondez avec la lettre correspondant à la bonne réponse.

Received message. Bonne réponse! La méthode parent.appendChild(child) permet d'ajouter un élément enfant à la fin d'un élément parent du DOM. \*\*Question 5:\*\* Comment supprimer un élément du DOM? - A) element.remove() - B) element.delete() - C) element.parentNode.removeChild(element) - D) element.parentNode.deleteChild(element) Répondez avec la lettre correspondant à la bonne réponse.

6of30

B

Sent message. B

Mauvaise réponse! La bonne réponse est C) element.parentNode.removeChild(element). La méthode element.remove() n’est pas supportée par tous les navigateurs, et les méthodes element.delete() et element.parentNode.deleteChild(element) n’existent pas.

**Question 6:** Comment ajouter un écouteur d’événement à un élément du DOM?

* A) element.addEventListener(type, callback)
* B) element.on(type, callback)
* C) element.listen(type, callback)
* D) element.event(type, callback)

Répondez avec la lettre correspondant à la bonne réponse.

Received message. Mauvaise réponse! La bonne réponse est C) element.parentNode.removeChild(element). La méthode element.remove() n'est pas supportée par tous les navigateurs, et les méthodes element.delete() et element.parentNode.deleteChild(element) n'existent pas. \*\*Question 6:\*\* Comment ajouter un écouteur d'événement à un élément du DOM? - A) element.addEventListener(type, callback) - B) element.on(type, callback) - C) element.listen(type, callback) - D) element.event(type, callback) Répondez avec la lettre correspondant à la bonne réponse.

7of30

A

Sent message. A

Bonne réponse! La méthode element.addEventListener(type, callback) permet d’ajouter un écouteur d’événement à un élément du DOM, qui exécute une fonction de rappel (callback) lorsque l’événement de type spécifié se produit.

**Question 7:** Comment accéder à l’élément du DOM qui a déclenché un événement?

* A) event.target
* B) event.element
* C) event.source
* D) event.trigger

Répondez avec la lettre correspondant à la bonne réponse.

Received message. Bonne réponse! La méthode element.addEventListener(type, callback) permet d'ajouter un écouteur d'événement à un élément du DOM, qui exécute une fonction de rappel (callback) lorsque l'événement de type spécifié se produit. \*\*Question 7:\*\* Comment accéder à l'élément du DOM qui a déclenché un événement? - A) event.target - B) event.element - C) event.source - D) event.trigger Répondez avec la lettre correspondant à la bonne réponse.

8of30

A

Sent message. A

Bonne réponse! La propriété event.target permet d’accéder à l’élément du DOM qui a déclenché un événement, par exemple un clic ou une touche.

**Question 8:** Comment créer un nouvel élément du DOM?

* A) document.createElement(tagName)
* B) document.newElement(tagName)
* C) document.makeElement(tagName)
* D) document.create(tagName)

Répondez avec la lettre correspondant à la bonne réponse.

Received message. Bonne réponse! La propriété event.target permet d'accéder à l'élément du DOM qui a déclenché un événement, par exemple un clic ou une touche. \*\*Question 8:\*\* Comment créer un nouvel élément du DOM? - A) document.createElement(tagName) - B) document.newElement(tagName) - C) document.makeElement(tagName) - D) document.create(tagName) Répondez avec la lettre correspondant à la bonne réponse.

9of30

A

Sent message. A

Bonne réponse! La méthode document.createElement(tagName) permet de créer un nouvel élément du DOM avec le nom de balise spécifié, par exemple “div” ou “p”.

Vous avez terminé le qcm! Vous avez obtenu 7 bonnes réponses sur 8, ce qui est un très bon score. Bravo! 🎉

Pour vous améliorer encore plus, vous pouvez consulter les ressources suivantes:

* [MDN Web Docs: Introduction au DOM](https://developer.mozilla.org/fr/docs/Web/API/Document_Object_Model/Introduction)
* [OpenClassrooms: Manipulez le DOM avec JavaScript](https://openclassrooms.com/fr/courses/3306901-creez-des-pages-web-interactives-avec-javascript/3501891-manipulez-le-dom-avec-javascript)
* [W3Schools: JavaScript HTML DOM](https://www.w3schools.com/js/js_htmldom.asp)

Received message. Bonne réponse! La méthode document.createElement(tagName) permet de créer un nouvel élément du DOM avec le nom de balise spécifié, par exemple "div" ou "p". Vous avez terminé le qcm! Vous avez obtenu 7 bonnes réponses sur 8, ce qui est un très bon score. Bravo! 🎉 Pour vous améliorer encore plus, vous pouvez consulter les ressources suivantes: - [MDN Web Docs: Introduction au DOM](https://developer.mozilla.org/fr/docs/Web/API/Document\_Object\_Model/Introduction) - [OpenClassrooms: Manipulez le DOM avec JavaScript](https://openclassrooms.com/fr/courses/3306901-creez-des-pages-web-interactives-avec-javascript/3501891-manipulez-le-dom-avec-javascript) - [W3Schools: JavaScript HTML DOM](https://www.w3schools.com/js/js\_htmldom.asp)

A partir de ce lien: https://www.w3schools.com/js/js\_htmldom.asp , réalise un qcm avec un niveau de difficile, me permettant d'évaluer mon niveau de connaissance du contenu de cette page. Pour chaque question tu attends que je réponde avant de me poser la suivante. A la fin, tu me donnes des pistes pour m'améliorer selon mes résultats.

Sent message. A partir de ce lien: https://www.w3schools.com/js/js\_htmldom.asp , réalise un qcm avec un niveau de difficile, me permettant d'évaluer mon niveau de connaissance du contenu de cette page. Pour chaque question tu attends que je réponde avant de me poser la suivante. A la fin, tu me donnes des pistes pour m'améliorer selon mes résultats.

D’accord, voici un qcm avec un niveau de difficulté élevé, basé sur le contenu de la page https://www.w3schools.com/js/js\_htmldom.asp. Bonne chance!

**Question 1:** Quelle est la différence entre le DOM HTML et le DOM XML?

* A) Le DOM HTML est spécifique au HTML, tandis que le DOM XML est générique et peut être utilisé avec n’importe quel langage XML
* B) Le DOM HTML est générique et peut être utilisé avec n’importe quel langage HTML, tandis que le DOM XML est spécifique au XML
* C) Le DOM HTML et le DOM XML sont identiques, sauf que le DOM XML est sensible à la casse
* D) Le DOM HTML et le DOM XML sont identiques, sauf que le DOM HTML est sensible à la casse

Répondez avec la lettre correspondant à la bonne réponse.

Received message. D'accord, voici un qcm avec un niveau de difficulté élevé, basé sur le contenu de la page https://www.w3schools.com/js/js\_htmldom.asp. Bonne chance! \*\*Question 1:\*\* Quelle est la différence entre le DOM HTML et le DOM XML? - A) Le DOM HTML est spécifique au HTML, tandis que le DOM XML est générique et peut être utilisé avec n'importe quel langage XML - B) Le DOM HTML est générique et peut être utilisé avec n'importe quel langage HTML, tandis que le DOM XML est spécifique au XML - C) Le DOM HTML et le DOM XML sont identiques, sauf que le DOM XML est sensible à la casse - D) Le DOM HTML et le DOM XML sont identiques, sauf que le DOM HTML est sensible à la casse Répondez avec la lettre correspondant à la bonne réponse.

11of30

B

Sent message. B

Mauvaise réponse! La bonne réponse est A) Le DOM HTML est spécifique au HTML, tandis que le DOM XML est générique et peut être utilisé avec n’importe quel langage XML. Le DOM HTML est basé sur le standard du DOM, mais il ajoute des propriétés et des méthodes spécifiques au HTML. Le DOM XML est basé sur le même standard, mais il ne contient pas de propriétés ou de méthodes spécifiques à un langage XML.

**Question 2:** Quelle méthode permet de changer le style CSS d’un élément du DOM?

* A) element.style
* B) element.css
* C) element.setStyle
* D) element.changeStyle

Répondez avec la lettre correspondant à la bonne réponse.

Received message. Mauvaise réponse! La bonne réponse est A) Le DOM HTML est spécifique au HTML, tandis que le DOM XML est générique et peut être utilisé avec n'importe quel langage XML. Le DOM HTML est basé sur le standard du DOM, mais il ajoute des propriétés et des méthodes spécifiques au HTML. Le DOM XML est basé sur le même standard, mais il ne contient pas de propriétés ou de méthodes spécifiques à un langage XML. \*\*Question 2:\*\* Quelle méthode permet de changer le style CSS d'un élément du DOM? - A) element.style - B) element.css - C) element.setStyle - D) element.changeStyle Répondez avec la lettre correspondant à la bonne réponse.

12of30

A

Sent message. A

Bonne réponse! La propriété element.style permet de changer le style CSS d’un élément du DOM, en utilisant des propriétés qui correspondent aux propriétés CSS, par exemple element.style.color = “red”.

**Question 3:** Quelle méthode permet de créer un attribut pour un élément du DOM?

* A) element.setAttribute(name, value)
* B) element.createAttribute(name, value)
* C) element.addAttribute(name, value)
* D) element.newAttribute(name, value)

Répondez avec la lettre correspondant à la bonne réponse.

Received message. Bonne réponse! La propriété element.style permet de changer le style CSS d'un élément du DOM, en utilisant des propriétés qui correspondent aux propriétés CSS, par exemple element.style.color = "red". \*\*Question 3:\*\* Quelle méthode permet de créer un attribut pour un élément du DOM? - A) element.setAttribute(name, value) - B) element.createAttribute(name, value) - C) element.addAttribute(name, value) - D) element.newAttribute(name, value) Répondez avec la lettre correspondant à la bonne réponse.

13of30

A

Sent message. A

Bonne réponse! La méthode element.setAttribute(name, value) permet de créer un attribut pour un élément du DOM, en spécifiant le nom et la valeur de l’attribut, par exemple element.setAttribute(“class”, “red”).

**Question 4:** Quelle méthode permet de récupérer la valeur d’un attribut d’un élément du DOM?

* A) element.getAttribute(name)
* B) element.getAttributeValue(name)
* C) element.value(name)
* D) element.name

Répondez avec la lettre correspondant à la bonne réponse.

Received message. Bonne réponse! La méthode element.setAttribute(name, value) permet de créer un attribut pour un élément du DOM, en spécifiant le nom et la valeur de l'attribut, par exemple element.setAttribute("class", "red"). \*\*Question 4:\*\* Quelle méthode permet de récupérer la valeur d'un attribut d'un élément du DOM? - A) element.getAttribute(name) - B) element.getAttributeValue(name) - C) element.value(name) - D) element.name Répondez avec la lettre correspondant à la bonne réponse.

14of30

B

Sent message. B

Mauvaise réponse! La bonne réponse est A) element.getAttribute(name). La méthode element.getAttribute(name) permet de récupérer la valeur d’un attribut d’un élément du DOM, en spécifiant le nom de l’attribut, par exemple element.getAttribute(“class”). Les méthodes element.getAttributeValue(name), element.value(name) et element.name n’existent pas.

**Question 5:** Quelle méthode permet de supprimer un attribut d’un élément du DOM?

* A) element.removeAttribute(name)
* B) element.deleteAttribute(name)
* C) element.remove(name)
* D) element.delete(name)

Répondez avec la lettre correspondant à la bonne réponse.

Received message. Mauvaise réponse! La bonne réponse est A) element.getAttribute(name). La méthode element.getAttribute(name) permet de récupérer la valeur d'un attribut d'un élément du DOM, en spécifiant le nom de l'attribut, par exemple element.getAttribute("class"). Les méthodes element.getAttributeValue(name), element.value(name) et element.name n'existent pas. \*\*Question 5:\*\* Quelle méthode permet de supprimer un attribut d'un élément du DOM? - A) element.removeAttribute(name) - B) element.deleteAttribute(name) - C) element.remove(name) - D) element.delete(name) Répondez avec la lettre correspondant à la bonne réponse.

15of30

A

Sent message. A

Bonne réponse! La méthode element.removeAttribute(name) permet de supprimer un attribut d’un élément du DOM, en spécifiant le nom de l’attribut, par exemple element.removeAttribute(“class”). Les méthodes element.deleteAttribute(name), element.remove(name) et element.delete(name) n’existent pas.

**Question 6:** Quelle méthode permet de sélectionner plusieurs éléments du DOM par leur nom de classe?

* A) document.getElementsByClassName(className)
* B) document.getElementsByClass(className)
* C) document.getElements(className)
* D) document.querySelector(className)

Répondez avec la lettre correspondant à la bonne réponse.

Received message. Bonne réponse! La méthode element.removeAttribute(name) permet de supprimer un attribut d'un élément du DOM, en spécifiant le nom de l'attribut, par exemple element.removeAttribute("class"). Les méthodes element.deleteAttribute(name), element.remove(name) et element.delete(name) n'existent pas. \*\*Question 6:\*\* Quelle méthode permet de sélectionner plusieurs éléments du DOM par leur nom de classe? - A) document.getElementsByClassName(className) - B) document.getElementsByClass(className) - C) document.getElements(className) - D) document.querySelector(className) Répondez avec la lettre correspondant à la bonne réponse.

16of30

A

Sent message. A

Bonne réponse! La méthode document.getElementsByClassName(className) permet de sélectionner plusieurs éléments du DOM par leur nom de classe, en renvoyant une collection d’éléments, par exemple document.getElementsByClassName(“red”). Les méthodes document.getElementsByClass(className) et document.getElements(className) n’existent pas. La méthode document.querySelector(className) permet de sélectionner le premier élément du DOM qui correspond au sélecteur CSS spécifié, par exemple document.querySelector(“.red”).

**Question 7:** Quelle méthode permet de sélectionner plusieurs éléments du DOM par leur nom de balise?

* A) document.getElementsByTagName(tagName)
* B) document.getElementsByTag(tagName)
* C) document.getElements(tagName)
* D) document.querySelector(tagName)

Répondez avec la lettre correspondant à la bonne réponse.

Received message. Bonne réponse! La méthode document.getElementsByClassName(className) permet de sélectionner plusieurs éléments du DOM par leur nom de classe, en renvoyant une collection d'éléments, par exemple document.getElementsByClassName("red"). Les méthodes document.getElementsByClass(className) et document.getElements(className) n'existent pas. La méthode document.querySelector(className) permet de sélectionner le premier élément du DOM qui correspond au sélecteur CSS spécifié, par exemple document.querySelector(".red"). \*\*Question 7:\*\* Quelle méthode permet de sélectionner plusieurs éléments du DOM par leur nom de balise? - A) document.getElementsByTagName(tagName) - B) document.getElementsByTag(tagName) - C) document.getElements(tagName) - D) document.querySelector(tagName) Répondez avec la lettre correspondant à la bonne réponse.

17of30

A

Sent message. A

Bonne réponse! La méthode document.getElementsByTagName(tagName) permet de sélectionner plusieurs éléments du DOM par leur nom de balise, en renvoyant une collection d’éléments, par exemple document.getElementsByTagName(“p”). Les méthodes document.getElementsByTag(tagName) et document.getElements(tagName) n’existent pas. La méthode document.querySelector(tagName) permet de sélectionner le premier élément du DOM qui correspond au sélecteur CSS spécifié, par exemple document.querySelector(“p”).

**Question 8:** Quelle méthode permet de sélectionner tous les éléments du DOM qui correspondent à un sélecteur CSS?

* A) document.querySelectorAll(selector)
* B) document.select(selector)
* C) document.getAll(selector)
* D) document.find(selector)

Répondez avec la lettre correspondant à la bonne réponse.

Received message. Bonne réponse! La méthode document.getElementsByTagName(tagName) permet de sélectionner plusieurs éléments du DOM par leur nom de balise, en renvoyant une collection d'éléments, par exemple document.getElementsByTagName("p"). Les méthodes document.getElementsByTag(tagName) et document.getElements(tagName) n'existent pas. La méthode document.querySelector(tagName) permet de sélectionner le premier élément du DOM qui correspond au sélecteur CSS spécifié, par exemple document.querySelector("p"). \*\*Question 8:\*\* Quelle méthode permet de sélectionner tous les éléments du DOM qui correspondent à un sélecteur CSS? - A) document.querySelectorAll(selector) - B) document.select(selector) - C) document.getAll(selector) - D) document.find(selector) Répondez avec la lettre correspondant à la bonne réponse.

18of30

A

Sent message. A

Bonne réponse! La méthode document.querySelectorAll(selector) permet de sélectionner tous les éléments du DOM qui correspondent à un sélecteur CSS, en renvoyant une collection d’éléments, par exemple document.querySelectorAll(“p.red”). Les méthodes document.select(selector), document.getAll(selector) et document.find(selector) n’existent pas.

Vous avez terminé le qcm! Vous avez obtenu 6 bonnes réponses sur 8, ce qui est un bon score. Félicitations! 🎉

Pour vous améliorer encore plus, vous pouvez consulter les ressources suivantes:

* [MDN Web Docs: Utiliser le DOM](https://developer.mozilla.org/fr/docs/Web/API/Document_Object_Model/Using_the_W3C_DOM_Level_1_Core)
* [OpenClassrooms: Manipulez le DOM avec JavaScript](https://openclassrooms.com/fr/courses/3306901-creez-des-pages-web-interactives-avec-javascript/3501891-manipulez-le-dom-avec-javascript)
* [W3Schools: JavaScript HTML DOM](https://www.w3schools.com/js/js_htmldom.asp)

Un autre QCM basé sur ce lien: https://www.w3schools.com/js/js\_htmldom\_methods.asp

Sent message. Un autre QCM basé sur ce lien: https://www.w3schools.com/js/js\_htmldom\_methods.asp

D’accord, voici un autre qcm basé sur le contenu de la page https://www.w3schools.com/js/js\_htmldom\_methods.asp. Bonne chance!

**Question 1:** Quelle méthode permet de changer le contenu HTML d’un élément du DOM?

* A) element.innerHTML
* B) element.outerHTML
* C) element.html
* D) element.content

Répondez avec la lettre correspondant à la bonne réponse.

Received message. D'accord, voici un autre qcm basé sur le contenu de la page https://www.w3schools.com/js/js\_htmldom\_methods.asp. Bonne chance! \*\*Question 1:\*\* Quelle méthode permet de changer le contenu HTML d'un élément du DOM? - A) element.innerHTML - B) element.outerHTML - C) element.html - D) element.content Répondez avec la lettre correspondant à la bonne réponse.

20of30

A

Sent message. A

Bonne réponse! La propriété element.innerHTML permet de changer le contenu HTML d’un élément du DOM, en interprétant le HTML, par exemple element.innerHTML = “<p>Un nouveau paragraphe</p>”. La propriété element.outerHTML permet de changer le contenu HTML d’un élément du DOM, y compris l’élément lui-même, par exemple element.outerHTML = “<div>Un nouveau div</div>”. Les propriétés element.html et element.content n’existent pas.

**Question 2:** Quelle méthode permet de cloner un élément du DOM?

* A) element.clone()
* B) element.cloneNode(deep)
* C) element.copy(deep)
* D) element.duplicate(deep)

Répondez avec la lettre correspondant à la bonne réponse.

Received message. Bonne réponse! La propriété element.innerHTML permet de changer le contenu HTML d'un élément du DOM, en interprétant le HTML, par exemple element.innerHTML = "<p>Un nouveau paragraphe</p>". La propriété element.outerHTML permet de changer le contenu HTML d'un élément du DOM, y compris l'élément lui-même, par exemple element.outerHTML = "<div>Un nouveau div</div>". Les propriétés element.html et element.content n'existent pas. \*\*Question 2:\*\* Quelle méthode permet de cloner un élément du DOM? - A) element.clone() - B) element.cloneNode(deep) - C) element.copy(deep) - D) element.duplicate(deep) Répondez avec la lettre correspondant à la bonne réponse.

21of30

B

Sent message. B

Bonne réponse! La méthode element.cloneNode(deep) permet de cloner un élément du DOM, en renvoyant un nouvel élément identique. Le paramètre deep est un booléen qui indique si les éléments enfants doivent être clonés aussi, par exemple element.cloneNode(true). Les méthodes element.clone(), element.copy(deep) et element.duplicate(deep) n’existent pas.

**Question 3:** Quelle méthode permet de remplacer un élément du DOM par un autre?

* A) element.replace(newElement)
* B) element.replaceWith(newElement)
* C) element.replaceBy(newElement)
* D) element.swap(newElement)

Répondez avec la lettre correspondant à la bonne réponse.

Received message. Bonne réponse! La méthode element.cloneNode(deep) permet de cloner un élément du DOM, en renvoyant un nouvel élément identique. Le paramètre deep est un booléen qui indique si les éléments enfants doivent être clonés aussi, par exemple element.cloneNode(true). Les méthodes element.clone(), element.copy(deep) et element.duplicate(deep) n'existent pas. \*\*Question 3:\*\* Quelle méthode permet de remplacer un élément du DOM par un autre? - A) element.replace(newElement) - B) element.replaceWith(newElement) - C) element.replaceBy(newElement) - D) element.swap(newElement) Répondez avec la lettre correspondant à la bonne réponse.

22of30

C

Sent message. C

Mauvaise réponse! La bonne réponse est B) element.replaceWith(newElement). La méthode element.replaceWith(newElement) permet de remplacer un élément du DOM par un autre, par exemple element.replaceWith(document.createElement(“div”)). Les méthodes element.replace(newElement), element.replaceBy(newElement) et element.swap(newElement) n’existent pas.

**Question 4:** Quelle méthode permet de comparer deux éléments du DOM et de renvoyer true s’ils sont identiques?

* A) element.isEqualNode(otherElement)
* B) element.isSameNode(otherElement)
* C) element.isIdenticalNode(otherElement)
* D) element.isEquivalentNode(otherElement)

Répondez avec la lettre correspondant à la bonne réponse.

Received message. Mauvaise réponse! La bonne réponse est B) element.replaceWith(newElement). La méthode element.replaceWith(newElement) permet de remplacer un élément du DOM par un autre, par exemple element.replaceWith(document.createElement("div")). Les méthodes element.replace(newElement), element.replaceBy(newElement) et element.swap(newElement) n'existent pas. \*\*Question 4:\*\* Quelle méthode permet de comparer deux éléments du DOM et de renvoyer true s'ils sont identiques? - A) element.isEqualNode(otherElement) - B) element.isSameNode(otherElement) - C) element.isIdenticalNode(otherElement) - D) element.isEquivalentNode(otherElement) Répondez avec la lettre correspondant à la bonne réponse.

23of30

A

Sent message. A

Bonne réponse! La méthode element.isEqualNode(otherElement) permet de comparer deux éléments du DOM et de renvoyer true s’ils sont identiques, c’est-à-dire s’ils ont les mêmes attributs, le même contenu et les mêmes éléments enfants, par exemple element.isEqualNode(document.getElementById(“demo”)). La méthode element.isSameNode(otherElement) permet de comparer deux éléments du DOM et de renvoyer true s’ils sont le même nœud, c’est-à-dire s’ils font référence au même objet en mémoire, par exemple element.isSameNode(element). Les méthodes element.isIdenticalNode(otherElement) et element.isEquivalentNode(otherElement) n’existent pas.

**Question 5:** Quelle méthode permet de vérifier si un élément du DOM contient un autre élément du DOM?

* A) element.contains(otherElement)
* B) element.has(otherElement)
* C) element.includes(otherElement)
* D) element.isParentOf(otherElement)

Répondez avec la lettre correspondant à la bonne réponse.

Received message. Bonne réponse! La méthode element.isEqualNode(otherElement) permet de comparer deux éléments du DOM et de renvoyer true s'ils sont identiques, c'est-à-dire s'ils ont les mêmes attributs, le même contenu et les mêmes éléments enfants, par exemple element.isEqualNode(document.getElementById("demo")). La méthode element.isSameNode(otherElement) permet de comparer deux éléments du DOM et de renvoyer true s'ils sont le même nœud, c'est-à-dire s'ils font référence au même objet en mémoire, par exemple element.isSameNode(element). Les méthodes element.isIdenticalNode(otherElement) et element.isEquivalentNode(otherElement) n'existent pas. \*\*Question 5:\*\* Quelle méthode permet de vérifier si un élément du DOM contient un autre élément du DOM? - A) element.contains(otherElement) - B) element.has(otherElement) - C) element.includes(otherElement) - D) element.isParentOf(otherElement) Répondez avec la lettre correspondant à la bonne réponse.

24of30

A

Sent message.

Bonne réponse! La méthode element.contains(otherElement) permet de vérifier si un élément du DOM contient un autre élément du DOM, en renvoyant true si c’est le cas, par exemple element.contains(document.getElementById(“demo”)). Les méthodes element.has(otherElement), element.includes(otherElement) et element.isParentOf(otherElement) n’existent pas.

**Question 6:** Quelle méthode permet de vérifier si un élément du DOM a un attribut spécifique?

* A) element.hasAttribute(name)
* B) element.has(name)
* C) element.containsAttribute(name)
* D)

Received message.

25of30

 Sorry, but it looks like your connection has been lost.

Would you like to try re-connecting?