

Introduction Bitcoin Core

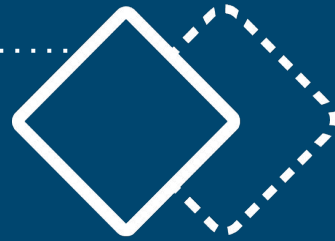


Défi #1: Outil d'analyse Bitcoin

ALYRA



Bitcoin



En 2008, Satoshi Nakamoto publie un [PDF](#) en ligne qui explique le fonctionnement d'une monnaie qu'il vient d'inventer

- Est un réseau de paiement novateur et une nouvelle forme d'argent.
- Technologie pair à pair
- Sans autorité centrale
- La gestion des transactions est prise en charge collectivement par le réseau

Comment ça marche ?

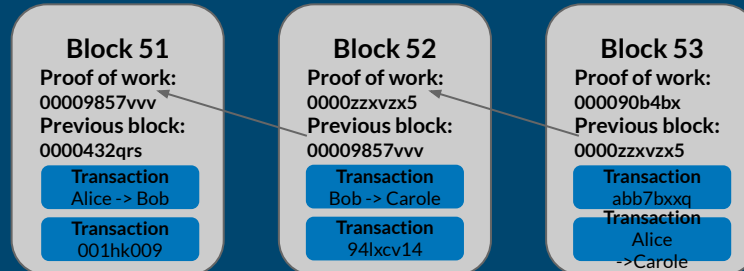


La Blockchain Bitcoin est un grand livre universel



Le 01/08/2009 à 16h : Alice paie 250 BTC à Bob
Le 03/08/2009 à 20h : Bob paie 30 BTC à Carole
Le 03/08/2009 à 22h : Alice paie 15 BTC à Carole

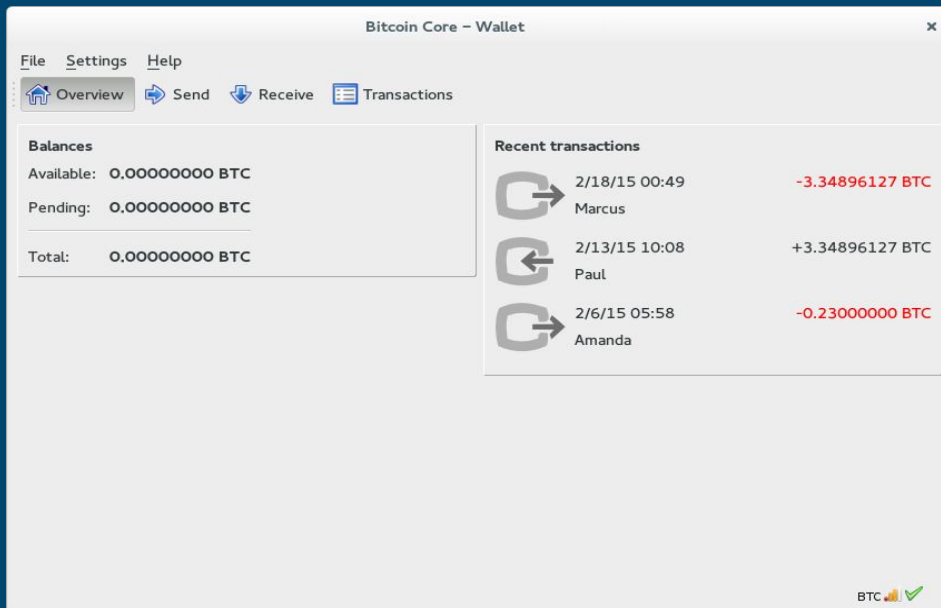
...



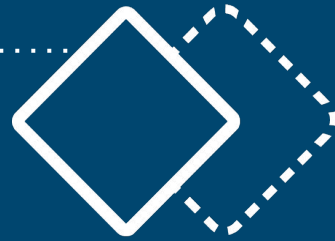
Bitcoin Core



Bitcoin Core est un des clients pair à pair de la crypto-monnaie Bitcoin.



Bitcoin Core

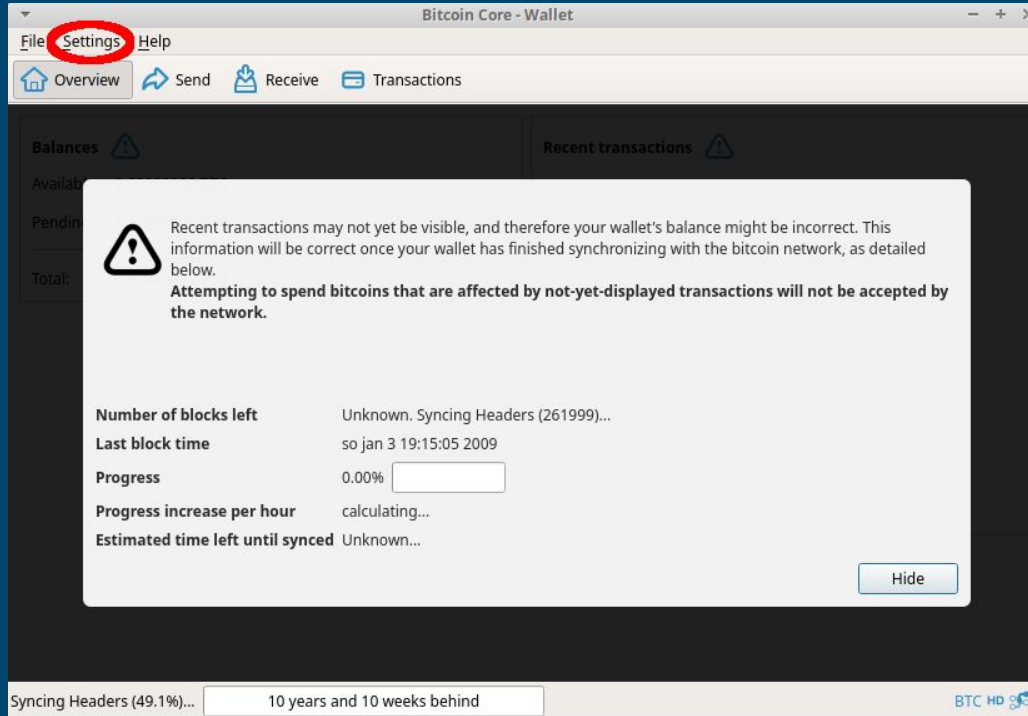


Bitcoin Core a 3 modes:

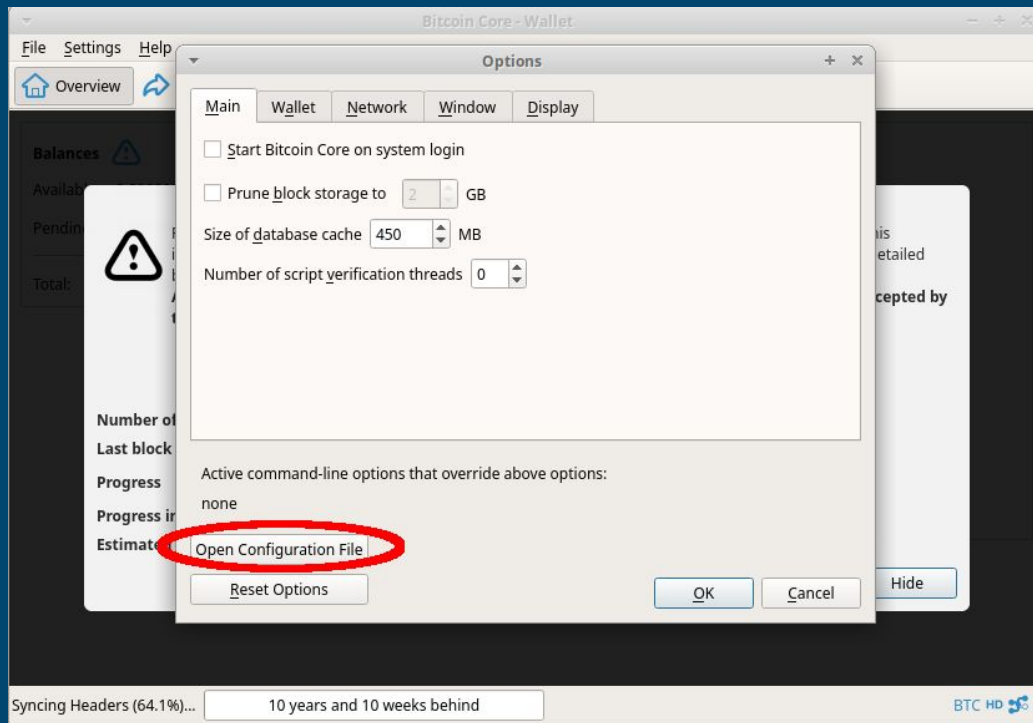
- Mainnet - est le réseau qui est utilisé comme version officielle, et il a de la valeur. Toutes les transactions réelles se font sur ce réseau.
- Testnet - un réseau qui a presque les mêmes règles que le Mainnet. Il a la découverte par les pairs, c'est-à-dire qu'il peut trouver des pairs sur le réseau testnet (comme sur le Mainnet) et un réseau peer-to-peer (p2p) le fait fonctionner.
- Regtest est une Blockchain privée qui a les mêmes règles et le même format d'adresse que testnet, mais il n'y a pas de réseau p2p global auquel se connecter.



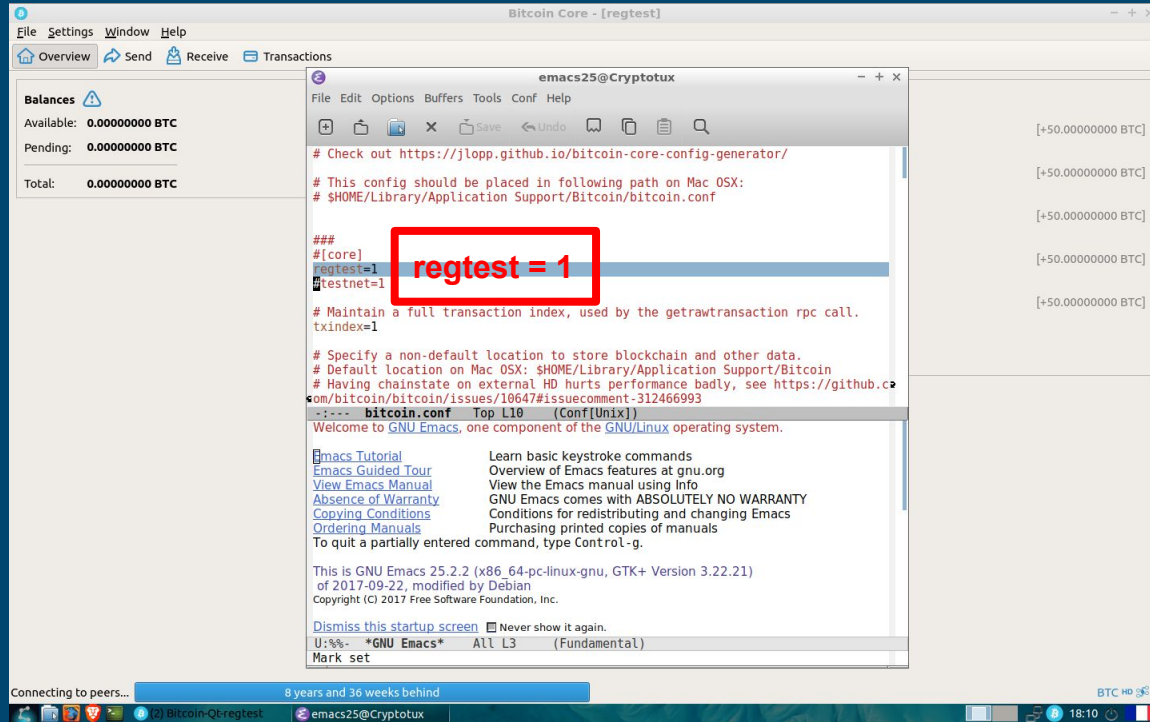
Comment passer sur regtest 1/3



Comment passer sur regtest 2/3



Comment passer sur regtest 3/3



The screenshot displays the Bitcoin Core desktop application and an Emacs editor window. On the left, the Bitcoin Core 'Overview' tab shows a balance of 0.00000000 BTC. The main window shows the Emacs editor with the `bitcoin.conf` file open. The configuration file contains several lines, with `regtest=1` highlighted by a red box and the text `regtest = 1` overlaid in red. The status bar at the bottom of the Emacs window shows 'U:%%- *GNU Emacs* All L3 (Fundamental)'. The system tray at the bottom indicates 'Connecting to peers... 8 years and 36 weeks behind' and shows the Bitcoin icon with a balance of 0.00000000 BTC.

Bitcoin Core - [regtest]

File Settings Window Help

Overview Send Receive Transactions

Balances ⓘ

Available: 0.00000000 BTC

Pending: 0.00000000 BTC

Total: 0.00000000 BTC

emacs25@Cryptotux

File Edit Options Buffers Tools Conf Help

+ Save Undo Find

Check out <https://jlopp.github.io/bitcoin-core-config-generator/>

This config should be placed in following path on Mac OSX:

\$HOME/Library/Application Support/Bitcoin/bitcoin.conf

###

#[core]

regtest=1

#testnet=1

Maintain a full transaction index, used by the getrawtransaction rpc call.

txindex=1

Specify a non-default location to store blockchain and other data.

Default location on Mac OSX: \$HOME/Library/Application Support/Bitcoin

Having chainstate on external HD hurts performance badly, see <https://github.com/bitcoin/bitcoin/issues/10647#issuecomment-312466993>

--- bitcoin.conf Top L10 (Conf[Unix])

Welcome to GNU Emacs, one component of the GNU/Linux operating system.

[Emacs Tutorial](#) Learn basic keystroke commands

[Emacs Guided Tour](#) Overview of Emacs features at gnu.org

[View Emacs Manual](#) View the Emacs manual using Info

[Absence of Warranty](#) GNU Emacs comes with ABSOLUTELY NO WARRANTY

[Copying Conditions](#) Conditions for redistributing and changing Emacs

[Ordering Manuals](#) Purchasing printed copies of manuals

To quit a partially entered command, type Control-g.

This is GNU Emacs 25.2.2 (x86_64-pc-linux-gnu, GTK+ Version 3.22.21)
of 2017-09-22, modified by Debian
Copyright (C) 2017 Free Software Foundation, Inc.

[Dismiss this startup screen](#) ☐ Never show it again.

U:%%- *GNU Emacs* All L3 (Fundamental)

Mark set

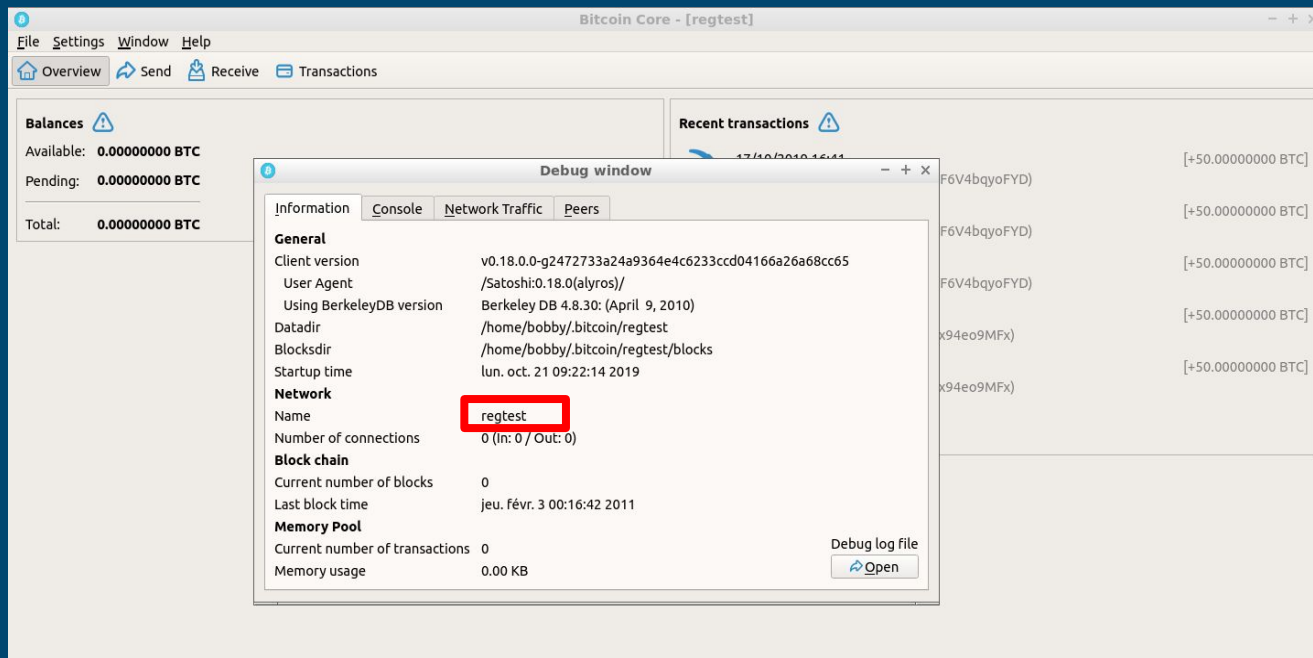
Connecting to peers... 8 years and 36 weeks behind

(2) Bitcoin-Qt-regtest emacs25@Cryptotux

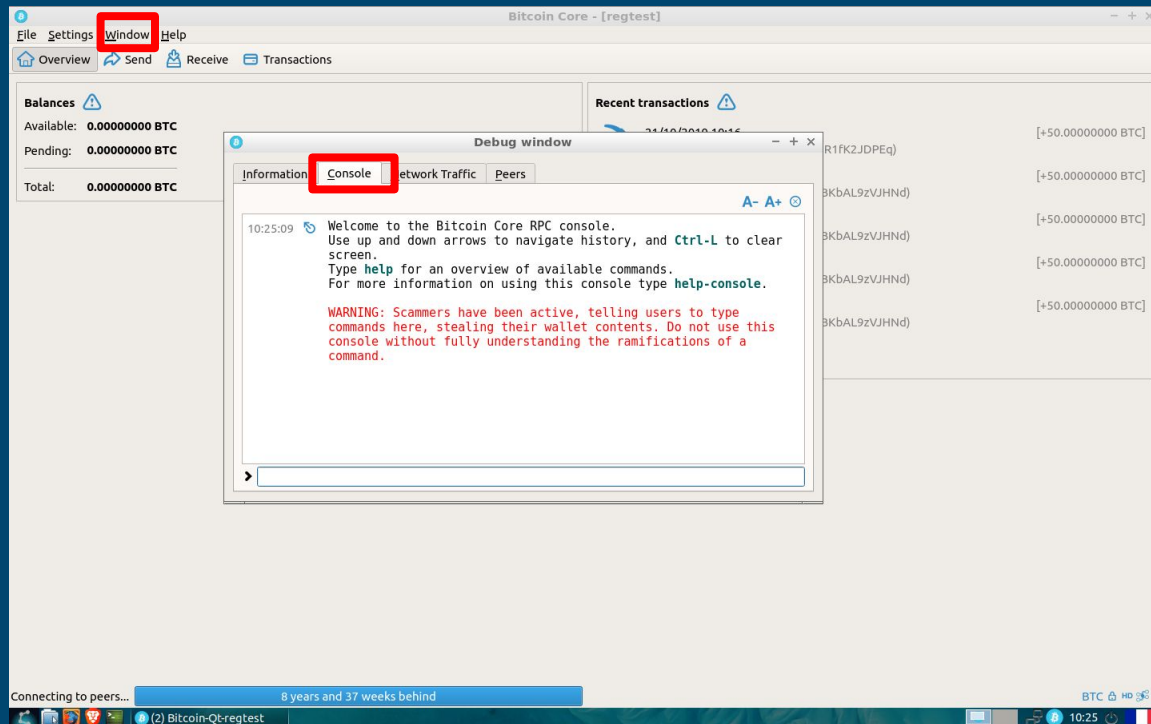
BTC 0.00000000

18:10

Comment vérifier le réseau sur Bitcoin Core



Console Bitcoin Core



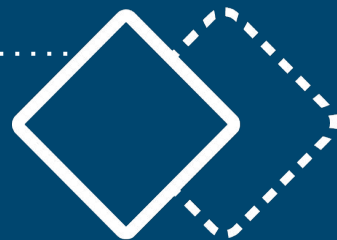
Démarrer Bitcoin Daemon (cli)

```
$ bitcoind
```

```
$ bitcoind -regtest -daemon
```



Quelques commandes (cli)



- Explorer la blockchain bitcoin
\$ bitcoin-cli -regtest getblockchaininfo
- Vérifier sa balance
\$ bitcoin-cli -regtest getbalance
- Générer une nouvelle adresse
\$ bitcoin-cli -regtest getnewaddress
- Miner un block
\$ bitcoin-cli -regtest generatetoaddress 1 "ADDRESS"
- Vérifier si le block a été miné
- Lister les transactions
\$ bitcoin-cli -regtest listtransactions

Défi #1



Objectif

- Faire un outil d'analyse de la blockchain Bitcoin

Sous forme

- Une page web
- Ou un utilitaire en ligne de commandes

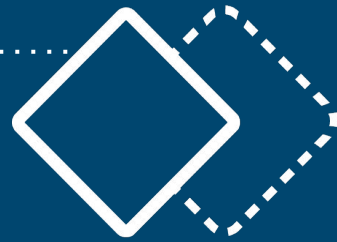
Rendu

- Fournir soit l'URL du dépôt (public)
- Soit directement le ou les fichiers correspondants

Equipe

- 5 équipes de 2 personnes
- 1 équipe de 3 personnes

Défi #1: Outil d'analyse Bitcoin

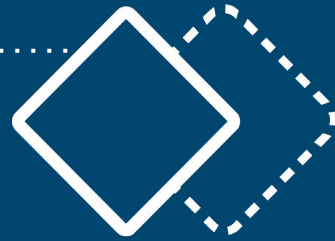


1. On lui donne des informations issues de la blockchain et il convertit les informations de façon explicite:
 - Hexadécimal -> décimal
 - Décimal -> hexadécimal
 - Hexadécimal little endian -> hexadécimal
 - varInt -> décimal
 -
2. Dans un deuxième temps, on lui fournit un bloc (sous format JSON ou hexadécimal) et il affiche les informations contenues.
Par exemple, vous pouvez prendre le bloc suivant :
<https://blockchain.info/rawblock/0000000000074a6f7e2d07cd8e5dd6dc8183993ee3b84666af499bc5b439d21b>

BONUS

- Récupérer les informations d'un noeud bitcoind directement en appelant l'interface JSON-RPC([https://en.bitcoin.it/wiki/API_reference_\(JSON-RPC\)](https://en.bitcoin.it/wiki/API_reference_(JSON-RPC))) et permettre la navigation.
- Défi additionnel, interpréter correctement la partie Script des transactions (opcodes et pile).

Système Hexadécimal

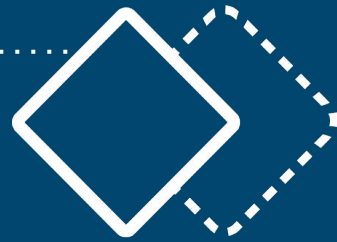


- Hexadécimal est un système de numération à base seize

Décimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hexa	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

- La liste des puissances de 16 vous sera utile pendant le processus de conversion.
 - $16^5 = 1\,048\,576$
 - $16^4 = 65\,536$
 - $16^3 = 4\,096$
 - $16^2 = 256$
 - $16^1 = 16$

Exercice : Convertir Hexa to Décimal

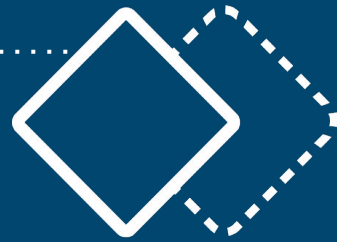


Convertir HEX -> Decimal

- Attribuez à chaque caractère la puissance qui lui correspond. Un nombre hexadécimal étant en base seize, c'est la place du caractère dans le nombre qui va déterminer la puissance de seize.
- Pour convertir en système décimal, multipliez chaque caractère (chiffre ou lettre) par la puissance de seize correspondante.
- Vous devez donc récrire l'hexa **C921** en décimal
 - $1 \times 16^0 = 1 \times 1$
 - $2 \times 16^1 = 2 \times 16$
 - $9 \times 16^2 = 9 \times 256$
 - $C \times 16^3 = C \times 4096 = 12 \times 4096$

=> Résultat = $1 + (2 \times 16) + (9 \times 256) + (12 \times 4096) = 51489$

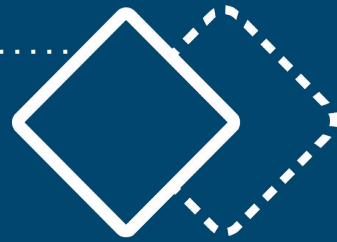
Exercice : Convertir Hexa to Décimal



```
function convertHexaToDecimal(h) {  
    decimal = parseInt(h,16)  
    return decimal  
}  
console.log(convertHexaToDecimal("C921"))
```

```
function hexToDec(hex) {  
    var result = 0, digitValue;  
    hex = hex.toLowerCase();  
    for (var i = 0; i < hex.length; i++) {  
        digitValue = '0123456789abcdefgh'.indexOf(hex[i]);  
        result = result * 16 + digitValue;  
    }  
    return result;  
}  
console.log(hexToDec('AC'));
```

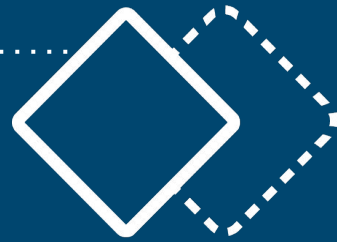
Exercice : Convertir Décimal to Hexa



Convertir 495 -> HEX

1. Trouvez la puissance de 16 la plus grande inférieure au nombre décimal.
Dans notre exemple, c'est « 256 ».
2. Divisez le nombre décimal par la puissance de 16. Arrêtez-vous au nombre entier et ignorez le reste du résultat après la virgule.
Dans notre exemple, $495 \div 256 = 1,93\dots$, mais seul le « 1 » nous intéresse ici.
Votre réponse est le premier chiffre du nombre hexadécimal.
3. Trouvez le reste. Cela vous permet de savoir ce qu'il se trouve à gauche du nombre que vous avez converti.
 - Multipliez la dernière réponse par le diviseur. Dans notre exemple : $1 \times 256 = 256$.
 - Soustrayez la réponse du dividende. $495 - 256 = 239$.
4. Divisez le reste par la prochaine puissance de 16 la plus élevée.
 - $239 \div 16 = 14$. Une fois de plus, vous ignorez tout ce qui se trouve après la virgule.

Exercice : Convertir Décimal to Hexa

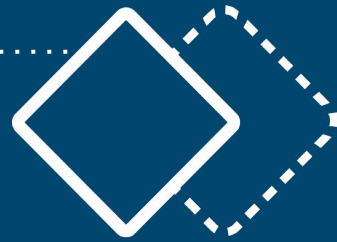


5. Trouvez une nouvelle fois le reste. Comme vous l'avez fait auparavant, multipliez la réponse par le diviseur, puis soustrayez la réponse du dividende. Vous devrez ensuite convertir le reste.
 - $14 \times 16 = 224$.
 - $239 - 224 = 15$, le reste est donc 15.
6. Recommencez jusqu'à ce que le reste soit inférieur à 16. Une fois que vous obtenez un reste entre 0 et 15, il est possible de le convertir directement par un seul chiffre hexadécimal. Écrivez ce dernier chiffre.
 - Le dernier « chiffre » de notre nombre hexadécimal est le 15, à la « première place ».

Résultat

(1) (14) (15) -> 1EF

Exercice : Convertir Décimal to Hexa



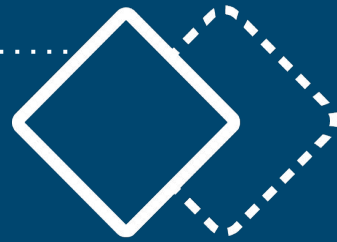
Vérifiez votre travail

Convertissez chacun des chiffres sous leur forme décimale, puis multipliez par la puissance de 16 de la position qu'ils occupent. Voici ce que vous devez faire pour notre exemple.

1EF → (1)(14)(15)

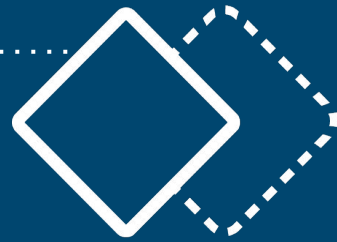
- De droite à gauche, 15 est en 16^0 , c'est-à-dire la première. $15 \times 1 = 15$.
- Le chiffre suivant sur la gauche est en $16^1 = 16$ e position. $14 \times 16 = 224$.
- Le chiffre suivant est en $16^2 = 256$ e position. $1 \times 256 = 256$.
- En les ajoutant tous ensemble, $256 + 224 + 15 = 495$, le nombre de départ.

Exercice : Convertir Décimal to Hexa



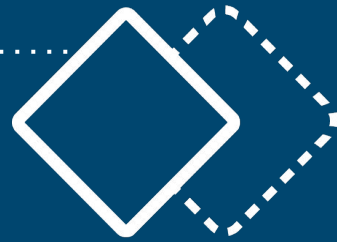
```
function convertDeciToHexa(h) {  
    hexa = h.toString(16)  
    if (h.length % 2 == 1) {  
        hexa = '0' + hexa  
    }  
    return hexa  
}  
  
console.log(convertDeciToHexa(172))
```

Exemple simple page web



```
<!DOCTYPE html>
<html>
  <head>
    <title>Page simple</title>
  </head>
  <body>
    Cette page est une
    page toute simple
  </body>
</html>
```

Exemple d'une page web avec fonction



```
<!DOCTYPE html>
<html>
  <head>
    <title>Bonjour</title>
  </head>
  <body>
    <h1 id="message"></h1>
    <button id="afficher" class="btn">Afficher</button>
    <script>
      function hello(){
        document.getElementById('message').innerHTML = 'Bonjour !'
      }
      document.getElementById('afficher').addEventListener('click', event => {
        hello()
      })
    </script>
  </body>
</html>
```

Exercices à rendre

Exercice 1.4.2 (pour Vendredi 25/10)

Exercice 2.2.1 (pour Vendredi 25/10)

Défi #1 (partie 1 pour Lundi 28/10)

Défi #1 (pour Vendredi 01/11)

