Distilling Your Message

Inbal Levi

What is std::vector?

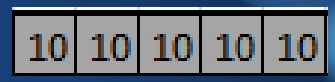
```
#include <vector>
#include <iostream>

using namespace std;

int main()
{
    // Create vector
    vector<int> v;
    // fill the array with 10 five times
    v.assign(5, 10);
    cout << "The vector elements are: ";
    for (int i = 0; i < v.size(); i++)
        cout << v.at(i) << " ";
    // inserts 15 to the last position
    v.push_back(15);
    int n = v.size();
    cout << "\nThe last element is: " << v.at(n -
1);
    // removes last element
    v.pop_back();
    // prints the vector
    cout << "\nThe vector elements are: ";
    for (int i = 0; i < v.size(); i++)
        cout << v.at(i) << " ";
    // inserts 5 at the beginning
    v.insert(v.begin(), 5);
    cout << "\nThe first element is: " << v.at(0);
    // removes the first element
    v.erase(v.begin());
    cout << "\nThe first element is: " << v.at(0);
    // inserts at the beginning
    v.emplace(v.begin(), 5);
    cout << "\nThe first element is: " << v.at(0);
    // Inserts 20 at the end
    v.emplace_back(20);
    n = v.size();
    cout << "\nThe last element is: " << v.at(n - 1);
    // erases the vector
    v.clear();
    cout << "\nVector size after erase(): " <<
v.size();
    // two vector to perform swap
    vector<int> v1, v2;
    v1.push_back(1);
    v1.push_back(2);
    cout << "\n\nVector 1: ";
    for (int i = 0; i < v1.size(); i++)
        cout << v1.at(i) << " ";
    cout << "\nVector 2: ";
    for (int i = 0; i < v2.size(); i++)
        cout << v2.at(i) << " ";
    // Swaps v1 and v2
    v1.swap(v2);
    cout << "\nAfter Swap \nVector 1: ";
    for (int i = 0; i < v1.size(); i++)
        cout << v1.at(i) << " ";
}
```

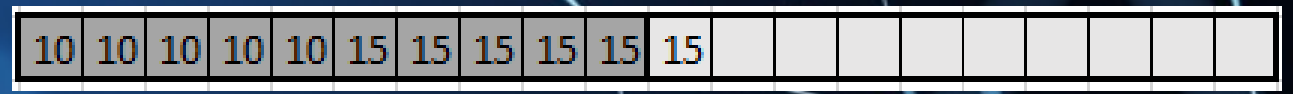
What is std::vector?

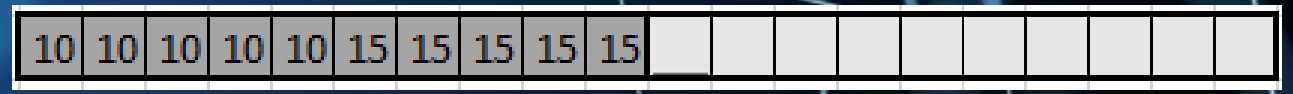
```cpp
#include <vector>
#include <iostream>

using namespace std;

int main()
{
    // Create vector
    vector<int> v;= {10, 10, 10, 10, 10};
    // fill the array with 10 five
times
    cout << "\nThe vector elements are: "
    print(0);
    for (int i = 0; i < v.size(); i++)
        cout << v.at(i) << " ";
    // inserts 15 to the last position
    v.push_back(15);        // {10, 10, 10, 10, 10,
15}
    int n = v.size();
    cout << "\nThe last element is: " << v.at(n -
1);
    // removes last element
    v.pop_back();           // {10, 10, 10, 10, 10}
    // prints the vector
    cout << "\nThe vector elements are: ";
    for (int i = 0; i < v.size(); i++)
        cout << v.at(i) << " ";
    // inserts 5 at the beginning
    v.insert(v.begin(), 5);
    cout << "\nThe first element is: " <<
   v.at(0);
}   // removes the first element
    v.erase(v.begin());
    cout << "\nThe first element is: " << v.at(0);
    // inserts at the beginning
    v.emplace(v.begin(), 5);
    cout << "\nThe first element is: " << v.at(0);
    // Inserts 20 at the end
    v.emplace_back(20);
    n = v.size();
    cout << "\nThe last element is: " << v.at(n - 1);
    // erases the vector
    v.clear();
    cout << "\nVector size after erase(): " <<
v.size();
    // two vector to perform swap
    vector<int> v1, v2;
    v1.push_back(1);
    v1.push_back(2);
    cout << "\n\nVector 1: ";
    for (int i = 0; i < v1.size(); i++)
        cout << v1.at(i) << " ";
    cout << "\nVector 2: ";
    for (int i = 0; i < v2.size(); i++)
        cout << v2.at(i) << " ";
    // Swaps v1 and v2
    v1.swap(v2);
    cout << "\nAfter Swap \nVector 1: ";
    for (int i = 0; i < v1.size(); i++)
        cout << v1.at(i) << " ";
}
```

What is std::vector?

```cpp
#include <vector>
#include <iostream>

using namespace std;

int main()
{

    vector<int> v= {10, 10, 10, 10, 10};


    print(v);       // {10, 10, 10, 10, 10}



    v.push_back(15);       // {10, 10, 10, 10, 10,
    15}

    (...)
    v.pop_back();          // {10, 10, 10, 10, 10}

}
```

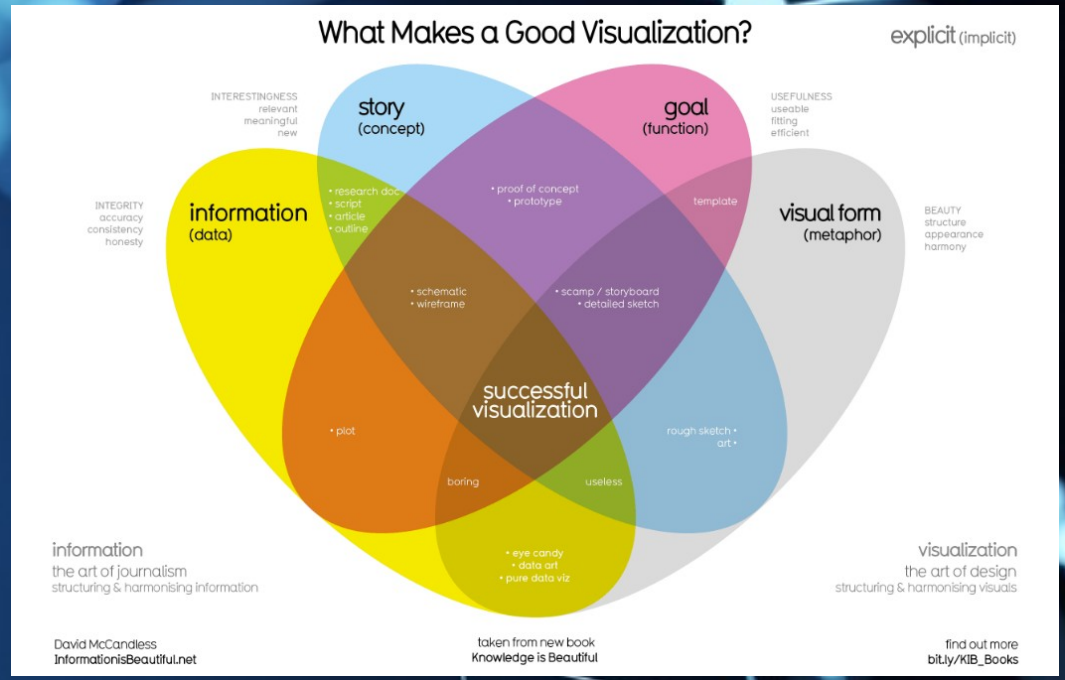






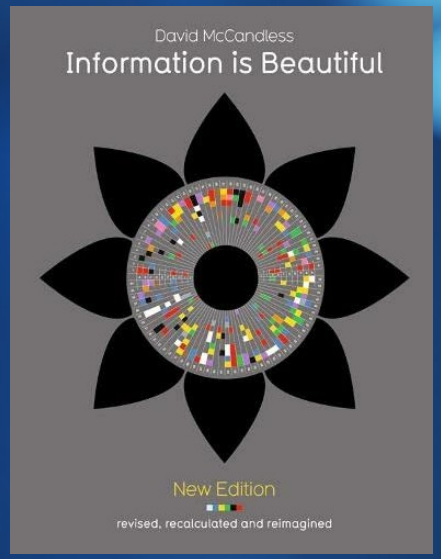
Data visualization

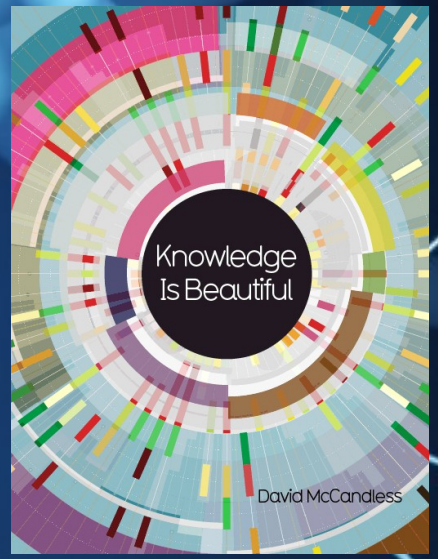
https://www.mygreatlearning.com/blog/introduction-to-data-visualisation-why-is-it-important/



https://www.informationisbeautiful.net/visualizations/what-makes-a-good-data-visualization/

Data visualization





- Create an experience for your audience
- Communicate with your audience
- Care about your audience

Creating a good talk

- Have a good structure
- Use visualization and minimalization in your talk
- Get perspectives, research topics related to what your talk is about
- Decide what are the take-aways
- Present your own views
- And also, make sure to bring all the equipment you need

Thank you,

And feel free to reach out!

linkdin/in/inballevi
twitter.com/InbaI_I
sinbal2lextra@gmail.com