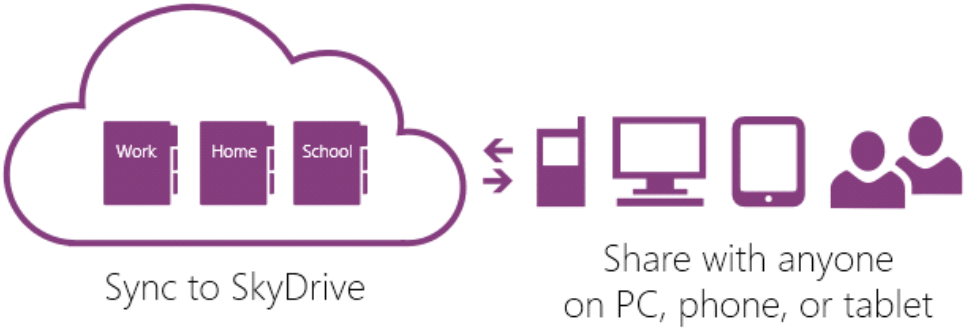
OneNote: one place for all of your notes



Watch the
2 minute video

1. Take notes anywhere on the page

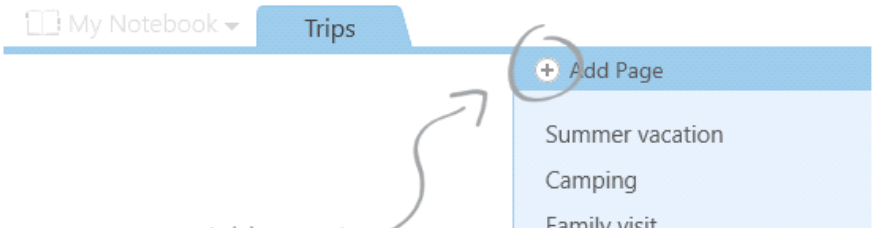
Write your name here

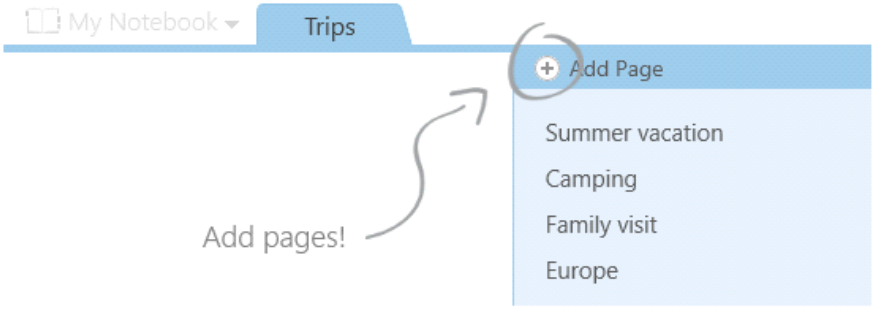
2. Get organized

You start with "My Notebook" - everything lives in here

Add sections for activities like:

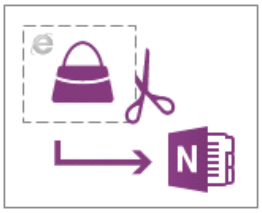
Add pages inside of each section:



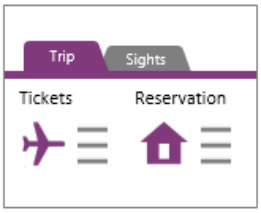


(Pages are over there)

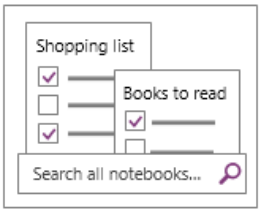
3. For more tips, check out 30 second videos



Clip from the web



Plan a trip with others



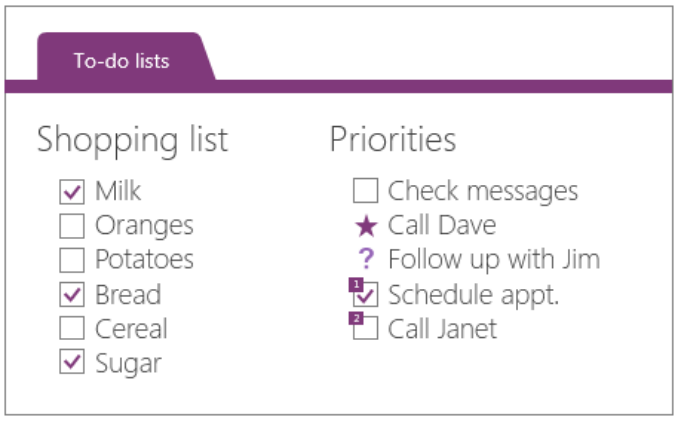
Search notes instantly

Write notes on slides

4. Create your first page

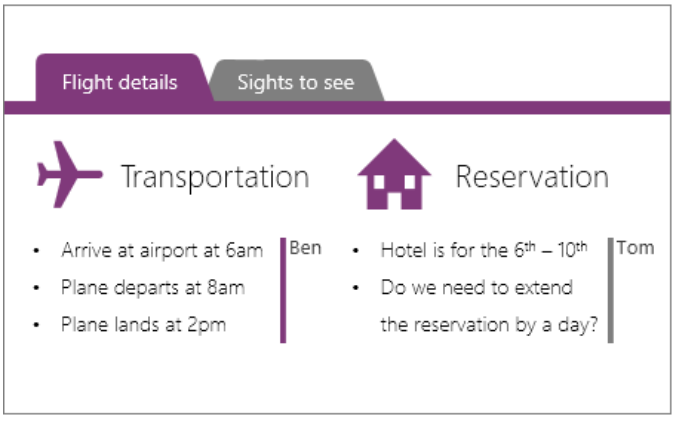
You're in the Quick Notes section - use it for random notes

OneNote Basics



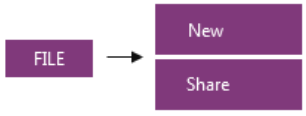
Remember everything

- Add Tags to any notes
- Make checklists and to-do lists
- Create your own custom tags



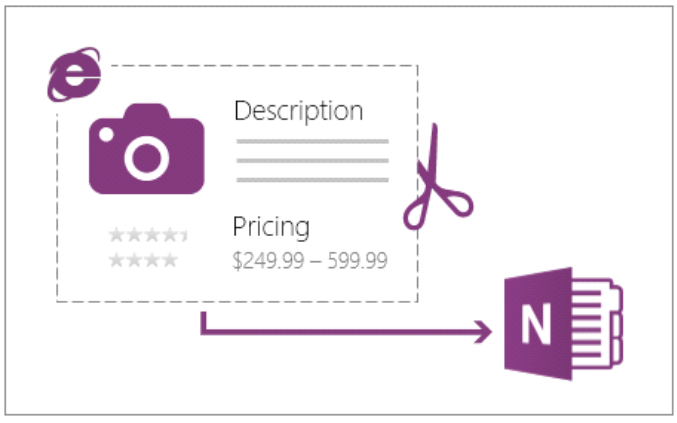
Collaborate with others

- Keep your notebooks on SkyDrive
- Share with friends and family
- Anyone can edit in a browser



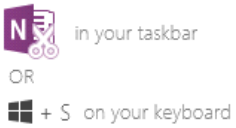
Keep everything in sync

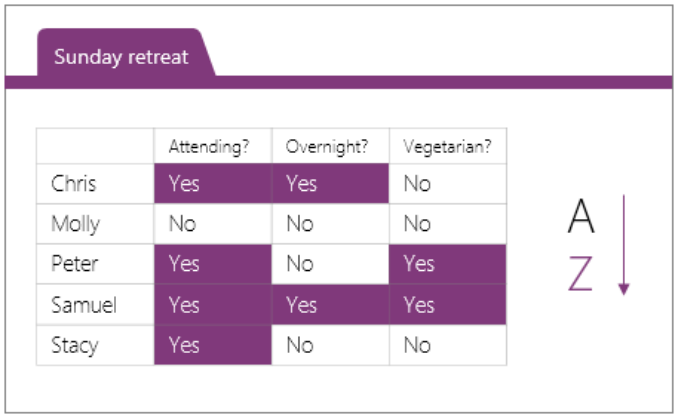
- People can edit pages at the same time
- Real-Time Sync on the same page
- Everything stored in the cloud
- Accessible from any device



Clip from the web

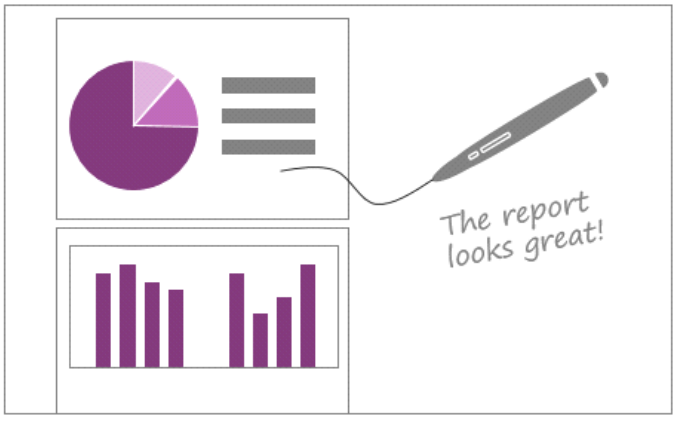
- Quickly clip anything on your screen
- Take screenshots of products online
- Save important news articles





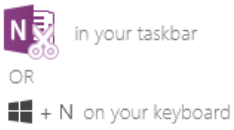
Organize with tables

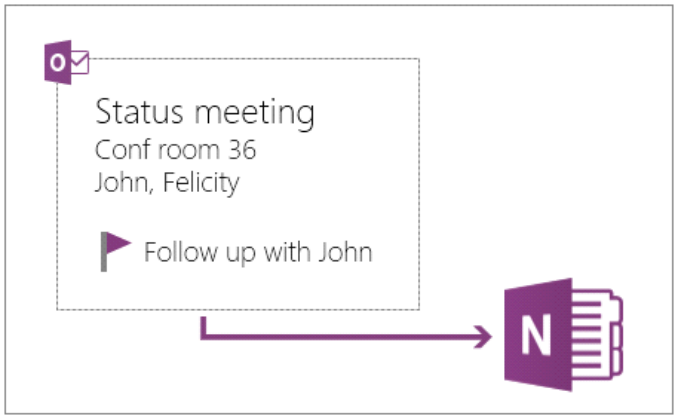
- Type, then press TAB to create a table
- Quickly sort and shade tables
- Convert tables to Excel spreadsheets



Write notes on slides

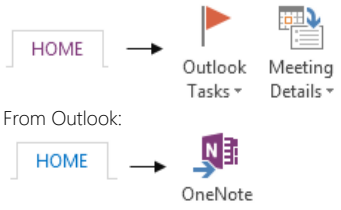
- Send PowerPoint or Word docs to OneNote
- Annotate with a stylus on your tablet
- Highlight and finger-paint

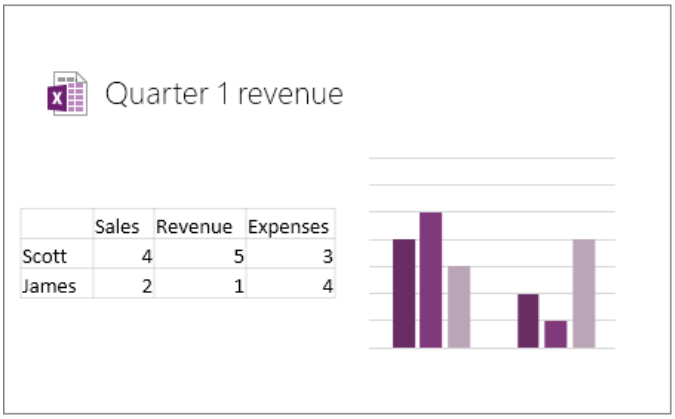




Integrate with Outlook

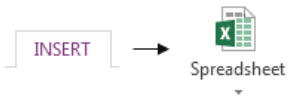
- Take notes on Outlook or Lync meetings
- Insert meeting details
- Add Outlook tasks from OneNote

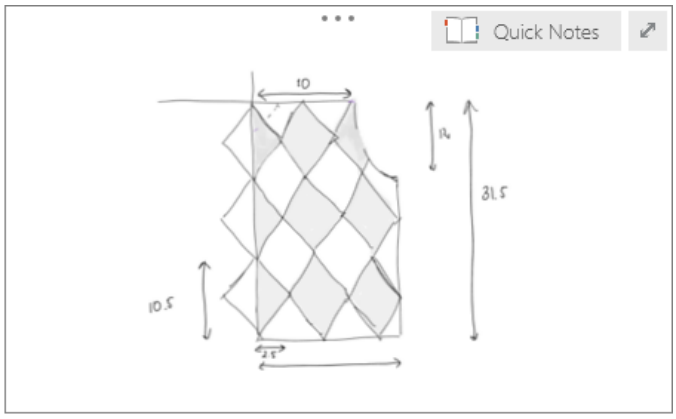




Add Excel spreadsheets

- Track finances, budgets, & more
- Preview updates on the page

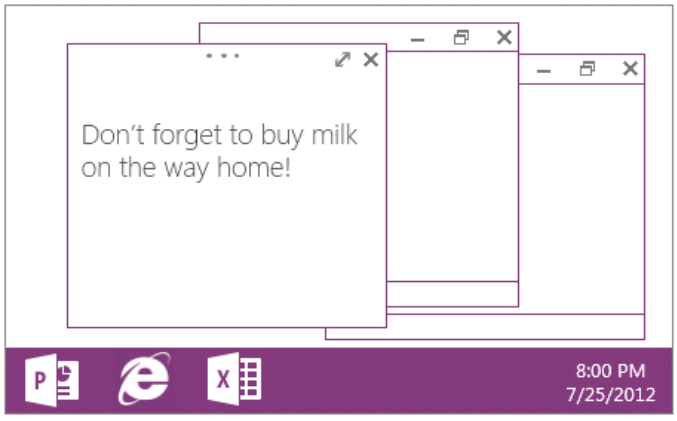




Brainstorm without clutter

- Hide everything but the essentials
- Extra space to focus on your notes

in the top corner of the page



Take quick notes

- Quickly jot down thoughts and ideas
- They go into your Quick Notes section

in your taskbar

OR

+ N on your keyboard

Generators

04 February 2014 09:43

```
function generateValues() {
console.log('Generating Value ...');

yield 1;

console.log('Generating Value 2 ...');

yield 2;

//throw new Error('Error Occurred!');

console.log('Generating Value 3 ...');

yield 3;
}

var generator = generateValues();

try {
while(true) {
console.log(generator.next());
}
} catch(error) {
if(error instanceof StopIteration)
console.log('Iteration Completed ...');
else
console.log('Error Occurred!');
} finally {
console.log('Operation Completed ...');

if(generator)
generator.close();
}

console.log();
```

Constants and Let Declarations

04 February 2014 09:58

Problems with var declarations

```
'use strict';

function problemsWithVar() {
var i = 0;

for(var j=i;j<10;j++) {
i++;
}

console.log('Value of i is : ' + i);
console.log('Value of j is : ' + j);

j++;

console.log('Value of j now is :' + j);
}

problemsWithVar();
```

Using Let in Javascript

```
'use strict';

function testingLets() {
let i = 10;

let(i = i+10) {
console.log('Value of i is : ' + i);
}

console.log('Value of i now is :' + i);

for(let j=0;j<i;j++)
console.log('Value of j is : ' + j);

alert(let (k=i+10) k);

// console.log('j now is : '+ j);

{
let i = i+10;
console.log('value of i is : ' + i);
}

console.log('value of i is : ' + i);
}

testingLets();
```

Constants in Javascript

```
'use strict';

const i = 10;

console.log('Value of i now is : ' + i);

 i = 20; // THROWS EXCEPTION (Strict Mode)
// DOES NOT THROW EXCEPTION (Non Strict Mode)

console.log('Value of i is : ' + i);

const j = i + 20;

console.log('value of j is : ' + j);
```

Understanding Destructuring Assignments

04 February 2014 10:13

Example #1

```
var a = 10;
var b = 20;

[b, a] = [a, b];

console.log('Value of b is : ' + b);
console.log('Value of a is : ' + a);
```

Example 2

```
function process() {
console.log('Processing Started ...');

return [10, 20, 30];
}

var [x, y, z] = process();

console.log('value of x is : ' + x);
console.log('value of y is : ' + y);
console.log('value of z is : ' + z);

var [i, , k] = process();

console.log('value of i is : ' + i);
console.log('value of k is : ' + k);

var [,,] = process();
```

Destructuring Assignments in Functions and For..each..in loop

```
var people =
[
{
firstName: 'Bill',
lastName: 'Gates',
location: {
country: 'USA'
}
},
{
firstName: 'Manmohan',
lastName: 'Singh',
location: {
country: 'India'
}
}
];

console.log('Javascript - Old way of iterating ...');

for each(let person in people) {
console.log(person.firstName + ', ' +
person.location.country);
}

console.log('Javascript 1.7 way of iterating ...');
```

```
for each(let {firstName: name, location: { country: place } } in people) {
console.log(name + ', ' + place);
}

function savePerson({firstName: name, location: { country : place }}) {
console.log(name + ', ' + place + ' ... Data Saved ...');
}

savePerson(people[0]);

console.log();
```

Array Comprehensions

04 February 2014 10:23

Learning Array Comprehensions and Expressions

```
function range(min, max) {
for(let x=min;x<max;x++)
yield x;
}

var filteredValues = [i for each(i in range(10,15))]

console.log(filteredValues);

var onlyEvenValues = [
i
for each(i in range(1, 10))
if(i % 2 == 0)];

console.log(onlyEvenValues);

function applyTaxProcessing (values) {
for each(let value in values) {
value = value * 0.9;

yield value;
}
};

try {
var generator = applyTaxProcessing(
[amount for each(amount in [100,200,300,400,500]) if(amount >= 200)]);

while(true) {
console.log(generator.next());
}
} catch(error if error instanceof StopIteration) {
console.log('Iteration Completed ...');
} catch(error) { console.log('Error Occurred ...' + error.message); }
finally {
if(generator)
generator.close();

console.log('Operation Completed!');
}
```

Generator Expressions

04 February 2014 10:40

```
try {
var values = [amount for each(amount in [100,200,300,400,500]) if(amount >= 200)];
var generator = (processedAmount * 0.9 for each(processedAmount in values));

while(true) {
console.log(generator.next());
}
} catch(error if error instanceof StopIteration) {
console.log('Iteration Completed ...');
} catch(error) { console.log('Error Occurred ...' + error.message); }
finally {
if(generator)
generator.close();

console.log('Operation Completed!');
}
```

Arrow Functions

04 February 2014 11:17

```
// New version of Javascript

function Execute(callback) {
var result = callback(10, 20);

return (x,y) => result + x + y;
}

console.log('Result is : ' +
Execute((a, b) => a * b) ( 10, 20 ));

// Old version of Javascript

function ExecuteX(callback) {
var result = callback(10,20);

return function(x, y) {
return result + x + y;
};
};

var ultimateResult = ExecuteX(function(a, b) {
return a * b;
});

console.log(ultimateResult(10, 20));
```

Another Example

```
function processOrder(orderId, orderDate, orderValue,
creditCardNumber,
   creditCardValidationCallback) {

   console.log('Order Processing Started ...');
   console.log('Details : ' + orderId + ', ' + orderDate.toString() + ', ' +
      orderValue + ', ' + creditCardNumber);

   if(creditCardValidationCallback && typeof
creditCardValidationCallback == 'function') {
      var status = creditCardValidationCallback(creditCardNumber);

      console.log('Validation Status : ' + status);
   }

   console.log('Order Processing Completed ...');
}

// processOrder('ORD10001', new Date(), 12000,
'VISA-1111-1212-1212',
//    function(creditCardNumber)
creditCardNumber.startsWith('VISA'));

processOrder('ORD10001', new Date(), 12000,
'VISA-1111-1212-1212',
   creditCardNumber => creditCardNumber.startsWith('VISA'));
```

Advanced Generators

04 February 2014 11:46

```
var genDefinition = () => (val for each(val in arguments));

try {
var generator = genDefinition(1,2,3,4,5);

while(true) {
console.log(generator.next());
}
} catch(error if error instanceof StopIteration) {
if(generator)
generator.close();
}
```

Iterator

04 February 2014 11:59

```
function Customer(id, name, address, creditLimit, activeStatus) {
this.id = id;
this.name = name;
this.address = address;
this.creditLimit = creditLimit;
this.activeStatus = activeStatus;
}

var customer = new Customer(1, 'Ramkumar', 'Bangalore', 12000, true);
var customerIterator = Iterator(customer, false);

for (let [key, value] in customerIterator) {
console.log(key + ', ' + value);
}

function Range(min, max) {
this.min = min;
this.max = max;
}

Range.prototype.__iterator__ = function() {
for(var value=this.min;value<this.max;value++)
yield value;
};

var range = new Range(10, 15);

for each(var item in range) {
console.log(item)
};
```

Objects and Properties

04 February 2014 12:06

```
'use strict';

var obj = Object.create(Object.prototype);

Object.defineProperty(obj, 'Id', {
value: 10,
enumerable: false,
writable: false,
configurable: false
});

Object.defineProperty(obj, 'Name', {
enumerable: true,
configurable: true,
get: function() { return 'Sony' },
set: function() { console.log('value has been set!'); }
});

console.log(obj.Id);

for (var property in obj)
console.log(property);

console.log(obj.Name);
```