1. OBJECT: Literal [ex](ex/objectliteral.htm), Function Declaration [ex](ex/functiondeclaration.htm), Constructor Function [ex](ex/constructorfunction.htm) 🡺 [vid](https://www.youtube.com/watch?v=RwT41El778A&list=PLFIM0718LjIUGpY8wmE41W7rTJo_3Y46-&index=1)
2. OBJECT.CREATE() [vid](https://www.youtube.com/watch?v=3pQ7Qpzl_pY&list=PLFIM0718LjIUGpY8wmE41W7rTJo_3Y46-&index=2) [ex](ex/objectcreate.htm)
3. PROTOTYPE [vid](https://www.youtube.com/watch?v=2CQhh_6xU3s&list=PLFIM0718LjIUGpY8wmE41W7rTJo_3Y46-&index=3) [ex1](ex/prototype.htm) [ex2](ex/prototype2.htm)
4. EXECUTING CONTEXT, HOISTING & SCOPE [vid](https://www.youtube.com/watch?v=8mZsm9ZQFdY&list=PLFIM0718LjIUGpY8wmE41W7rTJo_3Y46-&index=4) [ex](ex/contexhoistingscope.htm)
5. CLOSURE [vid](https://www.youtube.com/watch?v=jsW0taT36qU&list=PLFIM0718LjIUGpY8wmE41W7rTJo_3Y46-&index=5) [ex](ex/closure.htm)
6. VAR. LET & CONST -> var (function scope), let & const (block scope) -> jangan pernah gunakan var lagi [vid](https://www.youtube.com/watch?v=7HDgJScwIrI&list=PLFIM0718LjIUGpY8wmE41W7rTJo_3Y46-&index=6)
7. ARROW FUNCTION [vid](https://www.youtube.com/watch?v=C8U_3_SBk6s&list=PLFIM0718LjIUGpY8wmE41W7rTJo_3Y46-&index=7) [ex](ex/arrowfunction.htm)
8. THIS PADA ARROW FUNCTION [vid](https://www.youtube.com/watch?v=xVmUTO7O7GQ&list=PLFIM0718LjIUGpY8wmE41W7rTJo_3Y46-&index=8) [ex](ex/thisarrayfunction.htm)
9. HIGHER ORDER FUNCTION [vid](https://www.youtube.com/watch?v=sM880A3lS5I&list=PLFIM0718LjIUGpY8wmE41W7rTJo_3Y46-&index=9) [ex](ex/higherorderfunction.htm)
10. FILTER, MAP & REDUCE . METHOD CHAINING [vid](https://www.youtube.com/watch?v=a37wgCcBI4A&list=PLFIM0718LjIUGpY8wmE41W7rTJo_3Y46-&index=10) [vidLatihan](https://www.youtube.com/watch?v=V5-rQ1MCNSM&list=PLFIM0718LjIUGpY8wmE41W7rTJo_3Y46-&index=11) [ex](ex/filtermapreduce.htm) [exlatihan](ex/latihanfiltermapreduce.htm)
11. TEMPLATE LITERAL / TEMPLATE STRING [vid](https://www.youtube.com/watch?v=LywZF-xcfd8&list=PLFIM0718LjIUGpY8wmE41W7rTJo_3Y46-&index=12) [vidLatihan](https://www.youtube.com/watch?v=i4RIoT2tmpw&list=PLFIM0718LjIUGpY8wmE41W7rTJo_3Y46-&index=13) [ex](ex/templateliteral.htm) [exlatihan](ex/latihantemplateliteral.htm)
12. TAGGED TEMPLATE LITERAL [vid](https://www.youtube.com/watch?v=sbjkjjCcz8M&list=PLFIM0718LjIUGpY8wmE41W7rTJo_3Y46-&index=14) [ex](ex/taggedtemplates.htm)
13. DESTRUCTURING ASSIGNMENT [vid](https://www.youtube.com/watch?v=7f11bDxZSP0&list=PLFIM0718LjIUGpY8wmE41W7rTJo_3Y46-&index=15) [ex](ex/destructuringassignment.htm)
14. DESTRUCTURING FUNCTION [vid](https://www.youtube.com/watch?v=IcBNuBux9ag&list=PLFIM0718LjIUGpY8wmE41W7rTJo_3Y46-&index=16) [ex](ex/destructuringfunction.htm)
15. LOOPING FOR YANG BARU [vid](https://www.youtube.com/watch?v=LXOG9rHkYOo&list=PLFIM0718LjIUGpY8wmE41W7rTJo_3Y46-&index=17)

FOR .. OF => const mhs=[‘sandika’,’dodi’,’erik’]; for ( m of mhs){console.log(m)}; 🡺mhs.entries()???

(utk iterable object 1. String 2. Array 3. Arguments/NodeList 4. TyPed Array 5. Map 6. Set 7. User define i )

FOR .. IN => const mhs={nama:‘sandika’, umur: 31}; for ( m inf mhs){console.log(m)};

1. SPREAD OPERATOR : memecah iterables menjadi single elements [vid](https://www.youtube.com/watch?v=AT5hfOL1Ddk&list=PLFIM0718LjIUGpY8wmE41W7rTJo_3Y46-&index=18)

GABUNGKAN DUA ARRAY =>const dosen=[‘rahmadi’,’eko’]; const mhs=[‘tari’,’diah’]; const gabung=[…dosen,…mhs];

MENGCOPY ARRAY => const dosen=[‘rahmadi’,’eko’; const dosen1=[…dosen];

MENGUBAH NODELIST MENJADI ARRAY => limhs -> […limhs]

MEMECAH STRING => const nama=’RAHMADI’; const huruf = […nama.textContent];

1. REST PARAMETER => SPEREAD pada argument fungsi… [vid](https://www.youtube.com/watch?v=C0mPB-lyI1I&list=PLFIM0718LjIUGpY8wmE41W7rTJo_3Y46-&index=19)