1)prototypal inheritance-Tutaq ki,her hansı bir obyektin property ve metodlarını yaziriq.Əgər update elemek isdesek,sahib oldugumuz şeyləri birde kopyalamaq yerinə hemin obyekt uzerinde yeni obyekt yaradiriq.prototip bu obyeketde null ya da baska obyektden referans alir.Bu obyekte prototip deyilir.

2).Closure function : yaratdigimiz funksiyanin çölündeki variable almag bir basa munkun deyil.undefined qaytaracaq.Bunun üçün bizim ala biləcəyimiz bir funksiya yaradirig meselen get().Bu cür üsül clouser üsuludur.

3)typeof ya da intanceof vasitəsiylə.

4)Variable istiafade etmesek yazdığımız deyiskenler global object uzerinde yaradilir ve muxtelif yerlerdede isleyecek.

5)== eger degerleri eynidirse yeni typefolari ferqli olmasindan asli olmayalaraq true qaytarir.=== bu ise typeof-larina baxir ferlqidirse false qaytarir.

6)string,number,object

7)Degerlendirmir.Əgər basda string gəlirsə onnan sonra gelenleri ,meselen number olsa string kimi oxuyacaq.axirda string gelerse stringe catana qeder oldugu kimi oxuyacaq.

8)unobrostive javascript web site-in front-endə javascript elave etmeyin en yaxsi yoludur.Web Site-da javascript istifadə etmedikdə onnan istifade edilebilir etmek üçün unobtrusive javascript istifadə edilir.

9)-Undifined elave etdiyin deyisen var ama ne oldugunu bilmir.Yeni bir dəyər verilmiyib.

-Null isə deyisenin hec bir obyekte işarət etmediyini, müəyyən olmadigini bildirir.

10)

İf-Əgər şərt dogrudursa istifidə olunur

Else-səhfdirse istifade olunur

Else if-birinci sert sehfdirse ,test eləmək üçün yeni şərt yaratamak.

Switch-birden cox serti bildirmek ücün istifadə olunur.

11)NaN(not a number)-Deyisenin reqem olub olmadigini yoxluyur.

12)funksiya obyekti nedirse onu goturur ,hansi funksiyaya click eliyirkse onu götürür.

13)not-defined ümumiyyətlə cagirdigin bele bir deyisenin olmadigini bildirir.Undefined isə bele bi sey var ama her hansi bir deger vermedigini bildirir.

14)Hoisting funksiya sayesinde deyiseni declare etmeden evvelde istifade etmək olur.Yəni aftamatik olarak yuxariya daşıyır.

15)funksional və object-oriented.

16)funksional programming kodlari kisaltmag,oxunaxli etmek üçün istifadə olunur.

17)Classical inheritance-də siniflər valideynlerinden inheritance alir bulari her bir sinif üçün tək-tək yazmaq vaxt aparir.prototipal inheritance birden çox sinifden inheritance ala bilər bu daha optimaldır.

18)..

19)her inheritance lazım oldugda,obyektləri birden çox qaynaqdan götürmək istiyəndə.Modullarin ve ya funksional programming bariz bir hell tapa bilmiyəndə.

20)her obyektin prototip adlı ve baska obyekte baglanan bir iç xususiyyəti var.Zencir kimi. prototip zencirine olmuyan obyek sorgusu gonderdikde obyekti tapana qeder ve ya zencirin sonuna qeder axtarmaga devam eliyir.

21)object composition istifadə edsək daha sonra deyisiklik etmək rahat olar.injection ve setters köməyi ilə